Diversity collaboratively guided random drift particle swarm optimization

Li, C., Sun, J., Palade, V. & Li, L-W.

Original citation & hyperlink:
Li, C, Sun, J, Palade, V & Li, L-W 2021, 'Diversity collaboratively guided random drift particle swarm optimization', *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 9, pp. 2617–2638.
https://doi.org/10.1007/s13042-021-01345-1

DOI 10.1007/s13042-021-01345-1
ISSN 1868-8071
ESSN 1868-808X

Publisher: Springer

The final publication is available at Springer via http://dx.doi.org/10.1007/s13042-021-01345-1

Diversity collaboratively guided random drift particle swarm optimization

Chao Li[1], Jun Sun*[1], Vasile Palade[2], Li-Wei Li[1]

Address: [1] Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, No.1800, Lihu Avenue, Wuxi, Jiangsu 214122, PR China, [2] Faculty of Engineering, Environment and Computing, Coventry University, Priory Street, Coventry, CV1 5FB, UK

E-mail: Chao Li - lcmeteor@hotmail.com; Jun Sun* - sunjun_wx@hotmail.com; Vasile Palade - ab5839@coventry.ac.uk

*Corresponding author

Abstract: The random drift particle swarm optimization (RDPSO) algorithm is an effective random search technique inspired by the trajectory analysis of the canonical PSO and the free electron model in metal conductors placed in an external electric field. However, like other PSO variants, the RDPSO algorithm also inevitably encounters premature convergence when solving multimodal problems. To address this issue, this paper proposes a novel Diversity Collaboratively Guided (DCG) strategy for the RDPSO algorithm that enhances the search ability of the algorithm. In this strategy, two kinds of diversity measures are defined and modified in a collaborative manner. Specifically, the whole search process of the RDPSO is divided into three phases based on the changes in the two diversity measures. In each phase, different values are selected for the key parameters of the update equation in the RDPSO to make the particle swarm perform different search modes. Consequently, the improved RDPSO algorithm with the DCG strategy (DCG-RDPSO) can maintain its diversity dynamically at a certain level, and thus can search constantly without stagnation until the search process terminates. The performance evaluation of the proposed algorithm is done on the CEC-2013 benchmark suite, in comparison with several versions of RDPSO, different variants of PSO and several non-PSO evolutionary algorithms. Experimental results show that the proposed DCG strategy can significantly improve the performance and robustness of the RDPSO algorithm for most of the multimodal problems. Further experiments on economic dispatch problems also verify the effectiveness of the DCG strategy.

Keywords: Diversity guided strategy; Random drift particle swarm optimization algorithm; Multimodal optimization problems; Economic dispatch problems

1. Introduction

Particle swarm optimization (PSO) is a widely used random search optimization algorithm, which was inspired by the collective behavior of birds within a flock, and was first proposed by Eberhart and Kennedy (1995). By simulating the social learning and self-experience abilities of birds foraging, the particles in PSO exhibit similar search behaviors, including global search and local search, respectively. PSO is often used to solve non-continuous, complex and global optimization problems, since it is easy to use and can obtain generally good search performance with low computational cost.

Inspired by the trajectory analysis of the canonical PSO (CPSO) (Clerc and Kennedy 2002) and the free electron model in metal conductors placed in an external electric field (Omar 1975), Sun et al. (2013) proposed a novel variant of PSO, i.e. the random drift particle swarm optimization (RDPSO) algorithm. In this algorithm, it is assumed that each particle behaves like an electron moving in a metal conductor placed in an external electric field. The movement of the particle is thus the superposition of the thermal and the drift motions, which define the global search and the local search of the particle, respectively. The issues of particle's convergence behavior and parameter selection for the algorithm are addressed in detail in (Sun et al. 2015).

However, like CPSO and its variants, the RDPSO algorithm also inevitably encounter premature convergence, especially for multimodal optimization problems. This is because the local search plays a more important role for the algorithm in finding a good solution of the problem as the particle swarm converges during a limited number of iterations. Such kind of search mechanism is desirable for unimodal functions. In contrast, for multimodal problems, this may lead the algorithm to a local optimal or suboptimal solution when the search process finishes. Thus, how to enhance the global search ability of the algorithm, or put it more exactly, how to balance the exploration and exploitation of the algorithm in search process is what besets the researchers in the field of evolutionary computation and swarm intelligence.

During the last two decades, various kinds of strategies were implemented in PSO and its variants to tackle this problem. Niching (Sareni and Krahenbuhl 1998; Vitela and Castanos 2008) is a widely used strategy to improve the performance of the PSO algorithms. However, such methods require good specification of certain niching parameters according to the particular real problem to solve, in order to obtain good algorithmic performance (Brits et al. 2002; Li 2007; Liu et al. 2020). Therefore, the ring neighborhood topology (Kennedy 2006; Li 2009; Zou et al. 2020) was proposed to enhance the global search ability of the algorithm. Other strategies, such as hybridization with other evolutionary algorithms (Kao and Zahara 2008; Cao et al. 2018) and multi-population strategy (Zhan et al. 2013; Liu et al. 2017), also have their limitations. For example, some hybrid algorithms require more computing resources to perform different evolutionary processes (Kao and Zahara 2008; Cao et al. 2018), and most multi-population strategies cannot make the algorithm do finely local search at later stage of the search process (Liu et al. 2020).

Diversity-guided strategy is another widely used approach in the PSO algorithms to enhance the global search ability (Ursem 2002; Li et al. 2011; Wang et al. 2013; Janostik et al. 2016). However, as mentioned above, the local search ability is also very important in the search process, since the task of the algorithm is to find the global optimal solution of the problem, and, if that is not possible, a final result should be as good as possible. Thus, the key to solving multimodal problems effectively would be to properly balance the global and local search abilities of the algorithm. Explicitly, by controlling the diversity of the particle swarm, we can make the algorithm keep a certain animation so that its search ability can be maintained without stagnation over the search process. With this motivation in mind, this paper proposes a diversity collaboratively guided strategy to control the search behavior of the particles in RDPSO. In this strategy, the whole search process is divided into three phases according to the changes in two kinds of diversity measures. In each phase, different key parameter values are selected for the RDPSO to make the particle swarm perform different search modes. Such an improved RDPSO algorithm can dynamically maintain its diversity at a certain level during its search process. Therefore, it can keep searching the space as long as the termination condition is not satisfied. As such, the proposed algorithm can effectively solve high-dimensional and multimodal optimization problems. Evaluated on the CEC-2013 multimodal benchmark functions and some economic dispatch (ED) problems, the new Diversity Collaboratively Guided RDPSO (DCG-RDPSO) algorithm shows better performance than some RDPSO variants, PSO variants and some non-PSO evolutionary algorithms. This proves that the DCG strategy can definitely enhance the search ability and the robustness of RDPSO, and the DCG-RDPSO algorithm is a more reliable choice for solving multimodal problems than most of the compared algorithms.

The rest part of the paper is arranged as follows. In Section 2, the principles of PSO and RDPSO algorithms are introduced. In Section 3, the proposed diversity collaboratively guided strategy for the RDPSO is presented in detail. Section 4 presents the experimental results and comparative analysis between the proposed method and other algorithms. Some conclusions are given in Section 5.

2. Random drift particle swarm optimization

2.1 Particle swarm optimization

In a PSO with M individuals, each particle i $(1 \leq i \leq M)$ search in a N-dimensional real space, with its current position vector and velocity vector at the n^{th} iteration represented as $X_{i,n} = (X_{i,n}^1, X_{i,n}^2, \cdots, X_{i,n}^N)$ and $V_{i,n} = (V_{i,n}^1, V_{i,n}^2, \cdots, V_{i,n}^N)$, respectively. The particle moves according to the following equations:

$$V_{i,n+1}^j = V_{i,n}^j + c_1 r_{i,n}^j \left(P_{i,n}^j - X_{i,n}^j\right) + c_2 R_{i,n}^j \left(G_n^j - X_{i,n}^j\right) \quad (1)$$

$$X_{i,n+1}^j = X_{i,n}^j + V_{i,n+1}^j \tag{2}$$

for $i = 1,2\cdots,M$; $j = 1,2\cdots,N$, where c_1 and c_2 are known as the acceleration coefficients. The vector $P_{i,n} = (P_{i,n}^1, P_{i,n}^2, \cdots, P_{i,n}^N)$ is the best previous position found by particle i since initialization according to the fitness (or objective) function, and is called the personal best (*pbest*) position. The vector $G_n = (G_n^1, G_n^2, \cdots, G_n^N)$ is the best of all the *pbest* positions according to the fitness function and is called the global best (*gbest*) position. The *pbest* positions should be updated by comparing the fitness values of current positions and their own previous *pbest* positions. Generally, $r_{i,n}^j$, $R_{i,n}^j \sim U(0,1)$, that is, they are two different sequences of random numbers distributed uniformly on the interval (0, 1). The absolute value of the velocity of a particle in each dimension should be prevented from exceeding a given V_{max}. As equation (1) makes the particles in PSO search too globally and thus converge too slowly, Shi and Eberhart (1998) proposed the PSO algorithm with inertia weight, which is also known as the CPSO (Assareh et al. 2010). The update of velocity can be expressed as

$$V_{i,n+1}^j = w * V_{i,n}^j + c_1 r_{i,n}^j \left(P_{i,n}^j - X_{i,n}^j\right) + c_2 R_{i,n}^j \left(G_n^j - X_{i,n}^j\right) \tag{3}$$

where w is the inertia weight and is always set to linearly decrease from 0.9 to 0.4.

The trajectory analysis of CPSO (Clerc and Kennedy 2002) demonstrated that the convergence of the whole particle swarm may be achieved if each particle converges to its local focus $p_{i,n} = (p_{i,n}^1, p_{i,n}^2, \cdots, p_{i,n}^N)$ defined at the coordinates:

$$p_{i,n}^j = \frac{c_1 r_{i,n}^j P_{i,n}^j + c_2 R_{i,n}^j G_n^j}{c_1 r_{i,n}^j + c_2 R_{i,n}^j} \tag{4}$$

or

$$p_{i,n}^{j} = \phi_{i,n}^{j} P_{i,n}^{j} + \left(1 - \phi_{i,n}^{j}\right) G_{n}^{j}, \qquad \phi_{i,n}^{j} \sim U(0,1) \tag{5}$$

According to equation (5), $p_{i,n}$ is a random point uniformly distributed within the hyper-rectangle with $P_{i,n}$ and G_n being the two ends of its diagonal. Thus, the convergence behavior makes the particle search around the hyper-rectangle, which essentially reflects the local search of the particle. In equation (3), the first part on the right side with the inertia weight is called the "inertia part", which may lead the particle to fly away from $p_{i,n}$ or G_n, providing the necessary momentum for the particles to search globally in the search space; the last two parts on the right side are known as the "cognition part" and "social part", respectively, together leading the particle to move towards $p_{i,n}$. Therefore, the "inertia part" reflects the global search of the particle, while the superimposition of "cognition part" and "social part" reflects the local search of the particle.

2.2 Random drift particle swarm optimization

Similar to the electron moving in a metal conductor in an external electric field (Omar 1975), the movement of the particle in the RDPSO algorithm is thus the superposition of the thermal and the drift motions. Mathematically, the velocity of particle i in the jth dimension in RDPSO has two components:

$$V_{i,n+1}^{j} = VR_{i,n+1}^{j} + VD_{i,n+1}^{j} \tag{6}$$

where $VR_{i,n+1}^{j}$ and $VD_{i,n+1}^{j}$ are the random velocity and the drift velocity, corresponding to the thermal motion and the drift motion of the particle, respectively (Sun et al. 2015).

In RDPSO, the random velocity component $VR_{i,n+1}^{j}$ implements the global search. It replaces the 'inertia part' in the velocity equation of CPSO, which is the main difference between CPSO and RDPSO. It is assumed that the random velocity component should follow the Maxwell velocity distribution law (Kittel and Kroemer 1998). Thus, $VR_{i,n+1}^{j}$ essentially follows a normal distribution (i.e., Gaussian distribution) whose probability density function is given by

$$f_{VR_{i,n+1}^{j}}(v) = \frac{1}{\sqrt{2\pi}\sigma_{i,n+1}^{j}} e^{\frac{-v^2}{2(\sigma_{i,n+1}^{j})^2}} \tag{7}$$

where $\sigma_{i,n+1}^{j}$ is the standard deviation of the distribution. Using stochastic simulation, $VR_{i,n+1}^{j}$ can be expressed as

$$VR_{i,n+1}^{j} = \sigma_{i,n+1}^{j}\varphi_{i,n+1}^{j} \tag{8}$$

where $\varphi_{i,n+1}^{j}$ is a sequence of random numbers subject to standard normal position, i.e. $\varphi_{i,n+1}^{j} \sim N(0,1)$. $\sigma_{i,n+1}^{j}$ adopts an adaptive strategy expressed as

$$\sigma_{i,n+1}^{j} = \alpha|C_n^j - X_{i,n}^j| \tag{9}$$

where $\alpha > 0$ is a parameter in RDPSO, called the thermal coefficient. C_n is the mean best (*mbest*) position defined by the mean of the *pbes*t positions of all the particles, namely

$$C_n^j = (1/M)\sum_{i=1}^{M} P_{i,n}^j \tag{10}$$

As a result, equation (8) can be restated as

$$VR_{i,n+1}^{j} = \alpha|C_n^j - X_{i,n}^j|\varphi_{i,n+1}^{j} \tag{11}$$

On the other hand, the drift velocity component $VD_{i,n+1}^{j}$ is to achieve the local search of the particle. The original expression of $VD_{i,n+1}^{j}$ was first introduced in (Sun et al. 2010), which is just the combination of the “cognition part” and “social part” in CPSO, namely

$$VD_{i,n+1}^{j} = c_1 r_{i,n}^{j}\left(P_{i,n}^{j} - X_{i,n}^{j}\right) + c_2 R_{i,n}^{j}\left(G_{n}^{j} - X_{i,n}^{j}\right) \tag{12}$$

which is equivalent to

$$VD_{i,n+1}^{j} = (c_1 r_{i,n}^{j} + c_2 R_{i,n}^{j})\left(p_{i,n}^{j} - X_{i,n}^{j}\right) \tag{13}$$

where $p_{i,n}^{j}$ has been denoted in equation (5). However, equation (13) is not consistent with the free electron model, since $(c_1 r_{i,n}^{j} + c_2 R_{i,n}^{j})$ is a random number, making the movement of the particle randomized. Thus, a simple linear expression is adopted for $VD_{i,n+1}^{j}$ in this paper:

$$VD_{i,n+1}^{j} = \beta(p_{i,n}^{j} - X_{i,n}^{j}) \tag{14}$$

where $\beta > 0$ is another algorithmic parameter in RDPSO, called the drift coefficient. Equation (14) has a clear physical meaning, and in (Sun et al. 2015), it has been proved that when $0 < \beta < 2$, the expression of $VD_{i,n+1}^{j}$ in equation (14) can indeed guarantee the particle’s directional movement toward $p_{i,n}$ as an overall result.

With the above specification, a novel set of update equations can be obtained for the particle of the RDPSO algorithm:

$$V_{i,n+1}^{j} = \alpha|C_{n}^{j} - X_{i,n}^{j}|\varphi_{i,n+1}^{j} + \beta(p_{i,n}^{j} - X_{i,n}^{j}) \tag{15}$$

$$X_{i,n+1}^{j} = X_{i,n}^{j} + V_{i,n+1}^{j} \tag{16}$$

where the value of $V_{i,n}^{j}$ should also be restricted within $[-V_{max}, V_{max}]$.

Sun et al. (2015) highlighted that the RDPSO algorithm can obtain the average best performance when α is decreasing linearly from 0.9 to 0.3, with $\beta = 1.45$. In the following sections, the RDPSO algorithm with this parameter configuration of α and β is called the canonical RDPSO (CRDPSO).

3. Diversity collaborative guided RDPSO

In the RDPSO algorithm, there are essentially two swarms; that is, the *pbest swarm*, which is the set of all *pbest* positions in the swarm, and the *x swarm*, which is the set of all the particles' current positions in the particle swarm. Unlike PSO, the global search velocity and local search velocity components in the RDPSO are both related to the *pbest swarm* because of the definition of *mbest*. When all the *pbest* positions are distributed in a relatively large area, the velocities of most particles probably have large values, leading to a large distribution area for the current positions of all the particles. On the contrary, when all the *pbest* positions converge to a very small area, the current position of each particle always moves to a certain range of its own *pbest* position due to the dynamical behavior of the particle as analyzed in (Sun et al. 2015). This means the two particle swarms -- the *pbest swarm* and the *x swarm* -- are very close to each other. In this case, the velocities of particles should be of low value, which means the algorithm carries out more local search, in other words, searches in a small range. Note that narrowing the search range is necessary for the algorithm to improve the precision of the final solution. However, sharply reducing the search range and keeping a small-range search for a long period is not desirable since it can easily make the particles trapped into a local optimal solution. Unfortunately, in our preliminary experiments, it was found that sometimes the *pbest swarm* in the RDPSO converges too fast for some multimodal problems and then premature convergence is inevitable in these cases.

According to the above analysis, an obviously effective mechanism to tackle this problem is to control the change in the distribution area of *pbest swarm*. However, the *pbest* positions cannot be changed directly in case the previous search experience of particles would be eliminated. An effective way to achieve this goal is to modify the convergence speed of the *x swarm* to control the reduction rate of the distribution area of the *pbest swarm*. As a result, the appropriate distribution of the *pbest swarm* can affect the evolution of the *x swarm* as well. This is how these two swarms interact each other and work collaboratively.

Uresem (2002) first proposed the definition of diversity for PSO, which is measured by the average distance of the average point of the swarm and can be expressed as:

$$D(X_n) = \frac{1}{M \cdot A}\sum_{i=1}^{M}[\sum_{j=1}^{N}[X_{i,n}^{j} - \overline{X_n^J}]^2]^{1/2} = \frac{1}{M \cdot A}\sum_{i=1}^{M}\left|X_{i,n} - \overline{X_n}\right| \qquad (17)$$

where A is the diagonal length of search space, which represents the size of the search area. $\overline{X_n^J}$ is the j^{th} dimension of the mean of the all particles' positions.

According to this definition, $D(X_n)$ represents the proportion of the space distributed by the *x swarm* in the n^{th} iteration to the whole search space. The larger $D(X_n)$ is, the more dispersedly the particles distribute, and to some extent, the particle swarm is more diverse.

With this definition, we propose a novel diversity collaboratively guided (DCG) strategy for the RDPSO algorithm in this paper. In this strategy, the diversity of the *pbest swarm* (called *pbest diversity*) and the diversity of the *x swarm* (called *x diversity*) should be calculated in each iteration according to equation (17) during the whole search process. In addition, the above analysis reveals that the *pbest swarm* in the RDPSO should be prevented to converge too fast when solving multimodal problems, so that a baseline of the *pbest diversity* is set in this strategy to provide a desirable downward trend of the *pbest diversity*. Obviously, the aim of the DCG strategy is to make the *pbest diversity* decrease in accordance with the pre-set baseline by explicitly controlling the *x diversity* in the whole search process. To this end, one or both of the key parameters (α or/and β) in the velocity update equation of the RDPSO (equation (15)) is modified based on the relationship between the values of the two diversity measures and the baseline, so as to realize the divergence, global search or accelerated convergence of *x swarm*, and thus indirectly controlling the change of the *pbest diversity*. The details of the DCG strategy are described as follows.

A. The baseline of pbest diversity

As mentioned above, a decreasing baseline of *pbest diversity* is used in our proposed strategy, with a high value at the beginning, ensuring enough global search, and a low enough value at the later stage, making the particles search more locally. Our preliminary experiments showed that decreasing the baseline in a linear way is too slow while the one in an exponential way is too fast for multimodal problems. Thus, the polynomial form in terms of the iteration number is employed for decreasing baseline of *pbest diversity* in our strategy, which is expressed as:

$$B_n = (1 - n/n_{max})^c * (B_{start} - B_{end}) + B_{end} \quad (18)$$

where n represents the iteration number and n_{max} is the maximum number of iterations. c is a constant which determines the decline rate of the baseline. A low value of c implements more global search and less local search, while a high value means the opposite. Thus, c is obviously the key parameter of the proposed strategy, and its value can be set according to the situation of real problems. How to set the value of c for multimodal problems to generate generally good algorithmic performance will be discussed in the next section. B_{start} and B_{end} represent the start value and the end value of baseline, respectively. In this paper, B_{start} is set to the *pbest diversity* at the first iteration and B_{end} is set to a value obtained by a certain ratio times B_{start} as follows:

$$B_{start} = D(P_1) = \frac{1}{M \cdot A} \sum_{i=1}^{M} \left| P_{i,1} - \bar{P_1} \right| \tag{19}$$

$$B_{end} = eratio * B_{start} \tag{20}$$

where $eratio$ determines the final searching area for *x swarm*. In the later stage of the search process, a suitable small value of $eratio$ can make the *x swarm* keep searching in a relatively small area to lead the *gbest* position to improve quickly. Note that $eratio$ cannot be set to a too small value so that the particle swarm can have a certain global search ability even in the later stage of search. In this work, according to our preliminary experiments, $eratio = 1 \times 10^{-4}$ was set empirically, which can give the DCG strategy a good performance on multimodal problems.

With the given baseline of *pbest diversity*, the search process of the RDPSO algorithm can be controlled by comparing the baseline with the current *pbest diversity* and *x diversity*, so that the *pbest diversity* can be roughly reduced in line with the baseline.

B. Global search phase

When the *x diversity* is higher but the *pbest diversity* is lower than the baseline, the *pbest swarm* should converge slowly, waiting for the value of the baseline to decrease. In order to prevent a significant reduction of the *pbest diversity*, the *x swarm* cannot converge dramatically in this phase. Considering the relatively large difference between the size of the areas covered by the *pbest swarm* and the *x swarm*, we can assume that most of the particles are far away from their own *pbest* position in this situation. Thus, in this phase, $\alpha = 0.9$ and $\beta = 1.45$ are kept constant to make the particles implement global search within a certain area. Generally, this parameter setting makes both *x diversity* and *pbest diversity* decline moderately to maintain the global search ability of the algorithm at a certain level. The maximum value of α in the CRDPSO is selected in this phase, since except in the earlier stage of the search process, $\alpha = 0.9$ can effectively slow down the decline of the *x diversity*.

C. Divergence phase

When the current *pbest diversity* and *x diversity* are both lower than the current value of baseline, these two swarms should stop converging or converge more slowly than before to prevent the *pbest diversity* from dropping constantly to a too small value. To this end, a feasible way is to set α and/or β in the RDPSO algorithm to a high value to make the *x swarm* diverge until its diversity is larger than the baseline. Thus, in this phase, not only the *x diversity* experiences a rise, but also does the *pbest diversity* stop or slow down decreasing. However, if the parameters were not large enough, it would cost a lot of iterations in the divergence phase when the *x diversity* was much lower than the baseline. This is wasteful since, in this phase, there is not much chance for the particles to find a better *gbest* position, and moreover, the waste of many iterations in this phase would possibly slow down the whole convergence process. On the contrary, if too large values were given to the parameters, many particles would be easily fly beyond the search boundaries when its diversity value was large enough, and then the distribution of the particles would be related to handling of the out-of-bounds particles. This may result in the loss of searching experience of these particles. Thus, it is reasonable to change the parameters dynamically with the variation of diversity. Here, in this phase, the value of α is set to decrease with the rise of *x diversity* according to the following equation, but with fixed $\beta = 1.45$

$$\alpha = \alpha_0 / Dr(X_n), Dr(X_n) = D(X_n)/D(X_1) \tag{21}$$

where $D(X_1)$ is the *x diversity* in the first iteration, and $Dr(X_n)$ is the ratio of the diversity to $D(X_1)$ indicating how small the *x* diversity is. α_0 is set to 0.9, which is also the initial value of α in the CRDPSO. This can guarantee that the *x swarm* stop diverging when the $Dr(X_n)$ is near to 1, in case many particles reach the boundary of the search space. In equation (21), the reciprocal form is used for α so that the particles can still diverge to a relatively large area when the *x diversity* is low enough.

D. Accelerated convergence phase

Finally, if the *pbest diversity* is larger than the value of the baseline, the search process should switch to the accelerated convergence phase, regardless of whether the *x diversity* is larger or smaller than the baseline. Since the particle swarm experiences the divergence and global search phases in the search process, the convergence of the *pbest swarm* may be so slow that the decline rate of the *pbest diversity* is lower than that of the baseline. Consequently, the local search ability of the RDPSO may become weak and thus final fitness value of the *gbest* position may be of poor precision at the end of search. To deal with this problem, in the accelerated convergence phase, β is configured to decrease linearly from 1.45 to 1.05 in terms of iteration number, while α is set to be the same as in the CRDPSO, i.e. to decrease linearly from 0.9 to 0.3. The β value lower than that in the CRDPSO can definitely force the particles drift more quickly to the area closer to the *gbest* position, helping the particles update their *pbest* positions quickly. Consequently, the *pbest swarm* can be driven into a smaller region around the *gbest* position, resulting in that the *pbest diversity* can go down rapidly and finally catch up the baseline.

By combining the DCG strategy with the RDPSO model, we propose a novel variant of RDPSO, which is named as the Diversity Collaboratively Guided RDPSO (DCG-RDPSO) algorithm. Note that in the DCG-RDPSO, only except parameter c, all the other parameters mentioned in the DCG strategy do not need to be adjusted according to the particular situation of real problems. The procedure of the algorithm is outlined below.

Procedure of the DCG-RDPSO algorithm

Begin

Initialize the current position $X_{i,0}$ and the personal best positon $P_{i,0}$ of each particle, compute the mean best position C_0^j, evaluate their fitness values and find the global best position $G_0 = P_{g,0}$;

Pre-set c and $eratio$;

for $n = 1$ to n_{max} **do**

Calculate $D(X_n)$ and $D(P_n)$;

$B_{n,start} = D(P_1)$;
$B_{n,end} = eratio * D(P_1)$;
$B_n = (1 - n/n_{max})^c * (B_{n,start} - B_{n,end}) + B_{n,end}$;
if $D(P_n) < B_n$
if $D(X_n) < B_n$
$\alpha = 0.9/Dr(X_n), Dr(X_n) = D(X_n)/D(X_1)$;
$\beta = 1.45$;
end if
if $D(X_n) \geq B_n$
$\alpha = 0.9$;
$\beta = 1.45$;
end if
end if
if $D(P_n) \geq B_n$
$\alpha = 0.9 - (0.9 - 0.3) * n/n_{max}$;
$\beta = 1.45 - (1.45 - 1.05) * n/n_{max}$;
end if
for $i = 1$ to M **do**
for $j = 1$ to N **do**
$V_{i,n+1}^j = \alpha|X_{i,n}^j - C_n^j|\varphi_{i,n}^j + \beta(p_{i,n}^j - X_{i,n}^j)$;
$X_{i,n+1}^j = X_{i,n}^j + V_{i,n+1}^j$;
end for
Evaluate the objective function value $f(X_{i,n})$;
Update $P_{i,n}$ and G_n;
end for
end for
end

4. Experimental results and discussion

4.1 Benchmark and parameter setting

In order to find how to set the value of c and evaluate the effectiveness of the DCG-RDPSO algorithm in an empirical manner, the multimodal functions F_6 to F_{20} from the CEC-2013 benchmark suite (Liang et al. 2013) were employed for this purpose. The mathematical expressions and properties of the functions are described in detail in (Liang et al. 2013). Furthermore, in order to examine the performance of the DCG-RDPSO in practical applications, the proposed algorithm was used to solve the ED problems of three different real power systems. The definition of the ED problem and the characteristics of three systems are given in section 4.5.

The proposed algorithm was first compared with the RDPSO using different parameter configurations, including CRDPSO and the RDPSO with a linearly decreasing β (RDPSO-Dbeta), in order to show the effectiveness of the DCG strategy. Secondly, in order to show the good algorithmic performance of the proposed algorithm, the DCG-RDPSO was compared with some PSO algorithms, including CPSO (Clerc and Kennedy 2002), Fully Informed PSO (FIPS) (Mendes et al. 2004), Comprehensive Learning PSO (CLPSO) (Liang et al. 2006), Competitive and Cooperative PSO with Information Sharing Mechanism (CCPSO-ISM) (Li et al. 2015), Self Regulating PSO (SRPSO) (Tanweer et al. 2015), and PSO with Limited Information (LIPSO) (Du et al. 2015), and four non-PSO evolutionary algorithms, including artificial bee colony with distance-fitness-based neighbor search mechanism (DFnABC) (Cui et al. 2018), enhanced-based self-adaptive global-best harmony search (ESGHS) (Luo et al. 2019), memory-based global differential evolution (MGDE) (Zou et al. 2018), and self-adaptive real-coded genetic algorithm (SARGA) (Subbaraj et al. 2009). All the PSO algorithms mentioned above used fully connected topology, only except that the FIPS algorithm applied the ring topology. This is because according to (Mendes et al. 2004), the FIPS using fully connected topology could not reach criteria in all tested functions, while the one with the ring topology could achieve the highest success rate. Finally, all the aforementioned algorithms were employed to solve the ED problems of three practical power systems to verify the effectiveness of the proposed algorithm in real-world applications. Except the DCG-RDPSO algorithm, the parameter configurations of all the other compared algorithms are those recommended by the corresponding references, as shown in Table 1.

Table 1. Parameter configurations

Algorithm	Parameter Configurations
CRDPSO	$\alpha = 0.9 \rightarrow 0.3,\ \beta = 1.45$
RDPSO-Dbeta	$\alpha = 0.9 \rightarrow 0.3,\ \beta = 1.45 \rightarrow 1.05$
CPSO	$w = 0.9 \rightarrow 0.4,\ c_1 = c_2 = 2.0$
FIPS	$\chi = 0.7298,\ \sum c_i = 4.1$
CLPSO	$w = 0.9 \rightarrow 0.4,\ c = 1.49445,\ m = 7,\ Pc = 0.05 \sim 0.5$
CCPSO-ISM	$w = 0.6,\ c = 2.0,\ G = 5,\ P = 0.05$
SRPSO	$w_I = 1.05,\ w_F = 0.5,\ c_1 = c_2 = 1.49445,\ \eta = 1,\ \lambda = 0.5$
LIPSO	$\chi = 0.7298,\ \sum c_i = 4.1,\ W = 20$
DFnABC	$SN = 50,\ limit = SN \cdot D$ (D is the dimension of the problem)
ESGHS	$HMS = 100$
MGDE	$G_0 = G_{max}/100$ (G_{max} is the maximum iteration number)
SARGA	$p_c = 0.9,\ p_m = 0.01,\ \eta_c = 5,\ \eta_m = 1$

For most of the algorithms, the maximum iteration number was set to 30,000 and 3,000 for CEC-2013 functions and ED problems, respectively. The exception is the ESGHS, since in each iteration of ESGHS, only one candidate solution needs to be evaluated. Thus, the maximum iteration number of ESGHS was set to 3,000,000 and 300,000 for CEC-2013 functions and ED problems, respectively, to make the number of objective function evaluations of ESGHS equal to those of the other algorithms. All these algorithms terminated when the number of iterations reached the maximum iteration number. 100 particles were used for the PSO algorithms, MGDE and SARGA.

Each function in CEC-2013 and the objective function of each ED problem were optimized 51 times (same as the default setting in (Liang et al. 2013)) with respect to every tested algorithm. The boundary processing method for CEC-2013 functions used in all these algorithms was that if the component of a particle's current position in any dimension was beyond the search boundary, it should be replaced by a random value uniformly distributed within the defined search range. The constraint handling methods for the ED problems were introduced in section 4.5. With respect to the specific experimental setting for CEC-2013, the dimensionality for each tested benchmark function was 30, each dimension of all the particles was initialized randomly within the search range $[-100, 100]$, and each dimension of all particles' velocity was restricted in $[-100, 100]$. All the aforementioned experiments were implemented in the MATLAB programming environment (MATLAB R2018a) running on an Intel® i7-6850K 12-core 3.60 Giga-Hz CPU system with 128 Giga-Bytes memory.

4.2 Empirical studies on the parameter selection of the DCG strategy

As mentioned in the previous section, c is the key parameter in the DCG strategy. The value of c determines the decline speed of the baseline, which is closely related to the balance between the global search ability and local search ability of the algorithm. Therefore, in this section, a set of experiments on the different values of c were implemented. In each experiment, the value of c was an integer on the interval from 2 to 9. The mean final fitness values over 51 runs for F_6 to F_{20} yielded by the DCG-RDPSO algorithm are recorded in Table 2.

Table 2. Mean final fitness values and scores for the experiments with different c

values

c	2	3	4	5	6	7	8	9
F_6	**3.08E+01 (1.347)**	**2.90E+01 (0.975)**	**3.01E+01 (1.278)**	**3.07E+01 (1.441)**	**2.60E+01 (0.118)**	**2.56E+01 (0.000)**	**2.62E+01 (0.150)**	**3.34E+01 (1.829)**
F_7	**3.74E+00 (1.427)**	**4.09E+00 (1.583)**	**3.04E+00 (0.138)**	**3.24E+00 (0.476)**	**3.78E+00 (1.410)**	**2.96E+00 (0.000)**	**3.18E+00 (0.414)**	**3.23E+00 (0.510)**
F_8	**2.08E+01 (0.000)**	**2.08E+01 (0.956)**	**2.09E+01 (1.950)**	**2.09E+01 (1.556)**	**2.09E+01 (1.373)**	**2.08E+01 (0.796)**	**2.09E+01 (1.567)**	**2.09E+01 (1.358)**
F_9	1.20E+01 (2.377)	**1.15E+01 (1.615)**	**1.05E+01 (0.007)**	**1.12E+01 (1.196)**	**1.06E+01 (0.204)**	**1.08E+01 (0.488)**	**1.11E+01 (1.247)**	**1.05E+01 (0.000)**
F_{10}	6.15E-02 (3.373)	**4.47E-02 (0.056)**	**4.45E-02 (0.000)**	**4.84E-02 (0.966)**	**4.87E-02 (0.980)**	**4.47E-02 (0.056)**	**4.60E-02 (0.328)**	**4.80E-02 (0.794)**
F_{11}	3.04E+00 (5.327)	1.77E+00 (3.124)	1.44E+00 (2.256)	1.43E+00 (2.193)	**1.16E+00 (1.691)**	**1.02E+00 (1.438)**	**5.87E-01 (0.347)**	**4.69E-01 (0.000)**
F_{12}	**3.13E+01 (1.548)**	3.26E+01 (2.195)	3.28E+01 (2.287)	**3.12E+01 (1.437)**	**2.90E+01 (0.394)**	**3.08E+01 (1.210)**	**3.11E+01 (1.524)**	**2.83E+01 (0.000)**
F_{13}	7.17E+01 (2.925)	**6.91E+01 (1.905)**	**6.08E+01 (0.000)**	**6.23E+01 (0.355)**	**6.33E+01 (0.603)**	**6.31E+01 (0.553)**	**6.55E+01 (1.302)**	6.91E+01 (2.308)
F_{14}	5.32E+02 (3.808)	5.17E+02 (2.870)	4.25E+02 (2.275)	**3.32E+02 (0.000)**	**4.01E+02 (1.416)**	**3.84E+02 (1.213)**	**3.42E+02 (0.246)**	**3.72E+02 (0.731)**
F_{15}	5.13E+03 (3.365)	5.20E+03 (3.574)	**4.71E+03 (1.773)**	**4.57E+03 (1.255)**	4.80E+03 (2.149)	**4.24E+03 (0.000)**	**4.74E+03 (1.812)**	**4.64E+03 (1.432)**
F_{16}	**2.03E+00 (1.983)**	2.03E+00 (2.152)	**1.97E+00 (0.314)**	**1.95E+00 (0.000)**	**2.03E+00 (1.957)**	**1.97E+00 (0.451)**	**2.02E+00 (1.691)**	**1.99E+00 (0.941)**
F_{17}	4.57E+01 (13.668)	4.06E+01 (10.250)	3.60E+01 (2.086)	3.60E+01 (3.509)	**3.42E+01 (0.707)**	**3.50E+01 (1.840)**	**3.36E+01 (0.000)**	**3.46E+01 (1.295)**
F_{18}	**1.13E+02 (0.000)**	**1.29E+02 (1.921)**	**1.13E+02 (0.003)**	**1.28E+02 (1.919)**	1.30E+02 (2.206)	1.35E+02 (3.026)	1.37E+02 (3.281)	1.37E+02 (3.270)
F_{19}	2.18E+00 (7.962)	1.81E+00 (5.349))	1.62E+00 (2.683)	**1.49E+00 (0.902)**	**1.45E+00 (0.250)**	**1.50E+00 (1.073)**	**1.43E+00 (0.000)**	**1.52E+00 (0.976)**
F_{20}	**9.60E+00 (0.846)**	**9.60E+00 (0.800)**	**9.47E+00 (0.000)**	**9.69E+00 (1.397)**	**9.57E+00 (0.627)**	**9.68E+00 (1.319)**	**9.73E+00 (1.723)**	**9.53E+00 (0.374)**
Score	**7**	**8**	**10**	**13**	**13**	**14**	**14**	**13**

In Table 2, the result with an underline for each function is the best of mean final fitness values in all the experiments with different c values. As all the algorithms tested in our experiments are random search techniques, therefore, in this work, the t-test (Ruxton 2006) was used to determine whether there is a significant difference between the mean final fitness values obtained by two algorithms. In Table 2, the 51 results of each function with the same c value is considered as an independent group. Mathematically, the t-test established the problem statement by assuming a null hypothesis that the two means are the same between the two sample sets, and the t-value determine how likely the difference between the means occurred by chance. Since in this set of experiments, the variances of any two independent groups are different, the t-value should be calculated as (Ruxton 2006)

$$t = \frac{\mu_1 - \mu_2}{\sqrt{s_1^2/n_1 + s_2^2/n_2}} \tag{22}$$

where μ_1 and μ_2 are the mean values, s_1^2 and s_2^2 are the variances and n_1 and n_2 are number of samples associated with the first and second compared groups, respectively. In this work, the t-tests were undertaken at a 5% level of significance. Thus, according to the standard t tables, a t-value greater than 1.984 indicates that the null hypothesis can be rejected safely, that is, there is a significant difference between the two mean final fitness values; otherwise, the two mean final fitness values are regarded to be of no significant difference or to be equivalent in terms of statistical significance.

In Table 2, the best of mean final fitness value or equivalently best one in terms of t values for each function are marked bold. We scored in the last line the overall performance of the DCG-RDPSO with each different c value over all the benchmark functions, by counting the number of the bold results in each column. When $c = 7$ and $c = 8$, the algorithm has the most "best" results among all the cases. As a smaller c value can make the algorithm search more globally to avoid premature convergence, in this paper, $c = 7$ was thus used as recommend configuration for c and also was employed in all the following experiments.

4.3 Analysis of the DCG strategy

Table 3 records the mean final fitness values and the best final fitness values out of 51 runs on each benchmark function for the proposed algorithm and two versions of RDPSO with different parameter configurations, in order to verify the effectiveness of the proposed DCG strategy. With respect to the mean values, for each function, the best results among all the compared algorithms and the equivalently best ones are all in bold according to their corresponding t-values.

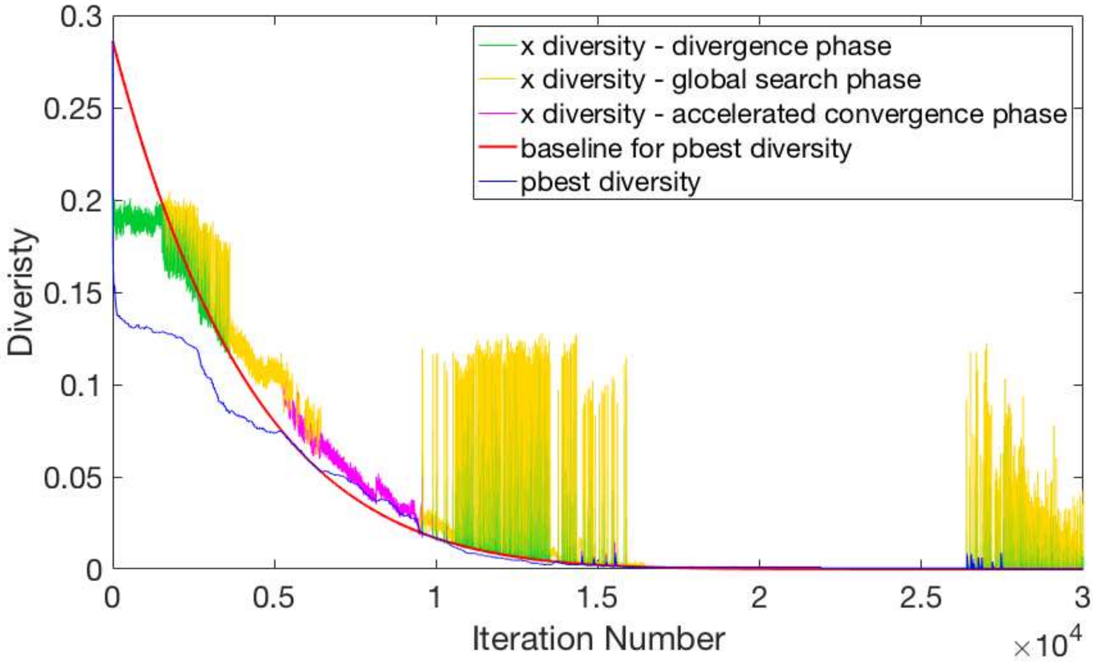
Table 3. Statistics results of three versions of RDPSO

Function		CRDPSO	RDPSO-Dbeta	DCG-RDPSO
F_6	Mean	3.59E+01	4.11E+01	**2.56E+01**
	(t-value)	(2.449)	(3.303)	**(0.000)**
	Best	1.05E+01	**6.64E-01**	1.46E+01
F_7	Mean	**3.81E+00**	4.31E+00	**2.96E+00**
	(t-value)	**(1.404)**	(2.125)	**(0.000)**
	Best	3.86E-01	3.94E-01	**1.96E-01**
F_8	Mean	**2.09E+01**	**2.09E+01**	**2.08E+01**
	(t-value)	**(1.065)**	**(1.302)**	**(0.000)**
	Best	2.07E+01	2.07E+01	**2.07E+01**
F_9	Mean	**1.10E+01**	**1.06E+01**	**1.08E+01**
	(t-value)	**(0.824)**	**(0.000)**	**(0.323)**
	Best	4.84E+00	4.35E+00	**4.17E+00**
F_{10}	Mean	**3.03E-02**	**3.42E-02**	4.47E-02
	(t-value)	**(0.000)**	**(0.917)**	(3.184)
	Best	8.10E-03	**3.48E-04**	7.40E-03
F_{11}	Mean	1.01E+01	9.97E+00	**1.02E+00**
	(t-value)	(16.993)	(19.373)	**(0.000)**
	Best	3.98E+00	5.97E+00	**2.76E-10**
F_{12}	Mean	4.58E+01	**3.14E+01**	**3.08E+01**
	(t-value)	(3.692)	**(0.279)**	**(0.000)**
	Best	1.31E+01	**1.10E+01**	1.20E+01
F_{13}	Mean	8.19E+01	7.55E+01	**6.31E+01**
	(t-value)	(4.242)	(2.759)	**(0.000)**
	Best	3.43E+01	2.55E+01	**1.50E+01**
F_{14}	Mean	8.87E+02	9.25E+02	**3.84E+02**
	(t-value)	(8.870)	(7.552)	**(0.000)**

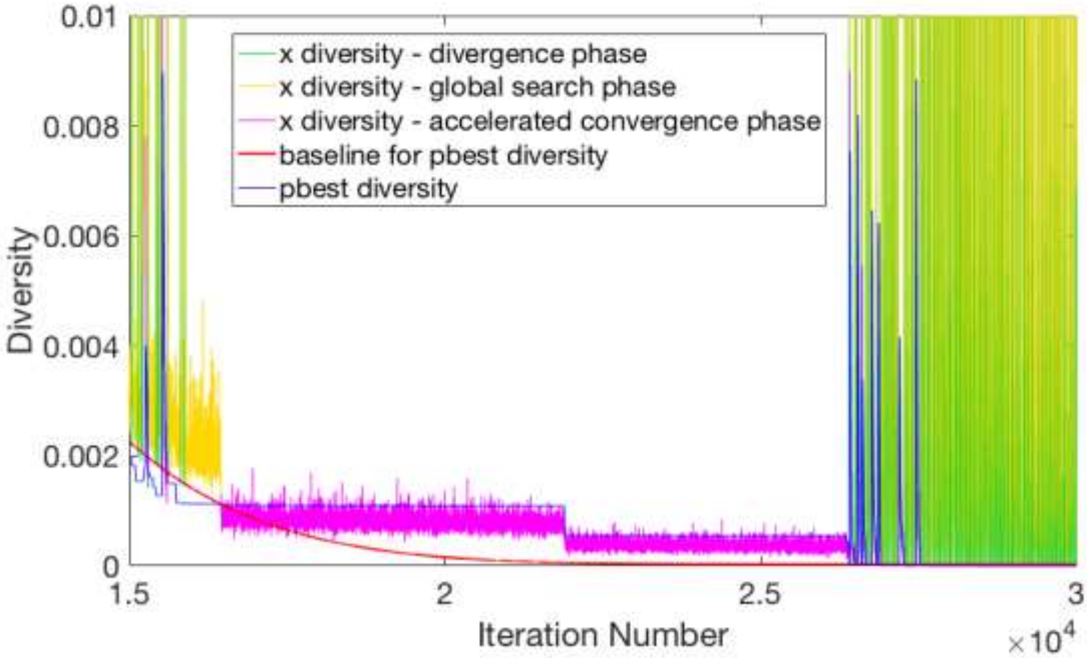
	Best	2.71E+02	3.09E+02	**2.52E+01**
F_{15}	Mean (t-value) Best	6.12E+03 (9.279) 3.04E+03	**4.63E+03** **(1.420)** 2.08E+03	**4.24E+03** **(0.000)** **1.71E+03**
F_{16}	Mean (t-value) Best	**1.99E+00** **(0.507)** 1.40E+00	**2.04E+00** **(1.527)** 1.32E+00	**1.97E+00** **(0.000)** **1.05E+00**
F_{17}	Mean (t-value) Best	4.78E+01 (6.549) 3.58E+01	4.43E+01 (10.883) 3.78E+01	**3.50E+01** **(0.000)** **3.06E+01**
F_{18}	Mean (t-value) Best	1.76E+02 (8.198) 1.52E+02	1.52E+02 (2.794) 6.91E+01	**1.35E+02** **(0.000)** **5.91E+01**
F_{19}	Mean (t-value) Best	2.63E+00 (8.919) 1.57E+00	2.66E+00 (13.768) 1.42E+00	**1.50E+00** **(0.000)** **1.01E+00**
F_{20}	Mean (t-value) Best	1.34E+01 (10.873) 8.12E+00	1.32E+01 (9.036) 8.22E+00	**9.68E+00** **(0.000)** **7.75E+00**

According to Table 3, it is obvious that with the help of DCG strategy, the DCG-RDPSO algorithm can get the best or equivalently best results for almost every tested multimodal function in terms of the mean value and the best value, except the mean value for F_{10}, and the best values for F_6, F_{10} and F_{12}, respectively. Moreover, according to the t-values, it can be found that in addition to F_8, F_9, F_{10} and F_{16}, the performance improvement of the DCG-RDPSO for all the other functions are very significant, which verifies the effectiveness of the DCG strategy.

To further demonstrate the performance of the operators in each phase of the DCG strategy, Fig.1 and Fig.2 illustrate the diversity curves as well as the baseline of F_{11} and F_{14} in one run as examples, respectively. The details of the second half of the two curves are enlarged to show more clearly the later stage of the search process. For the *x diversity*, different colors were used to mark different phases of the DCG strategy in these two figures.

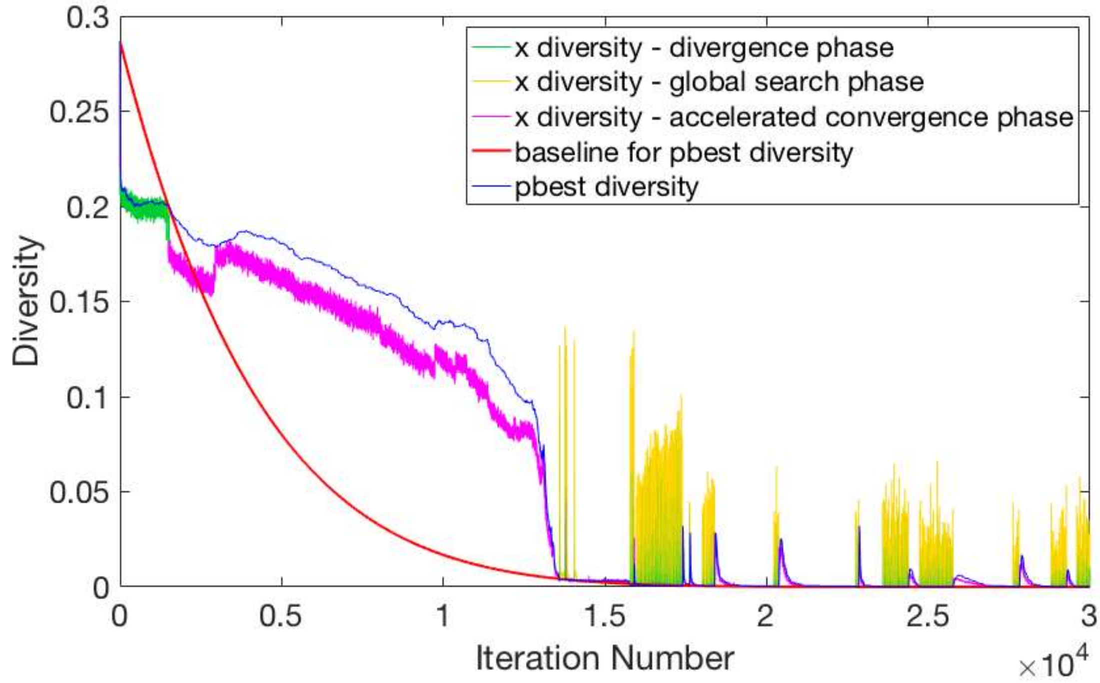


(a)

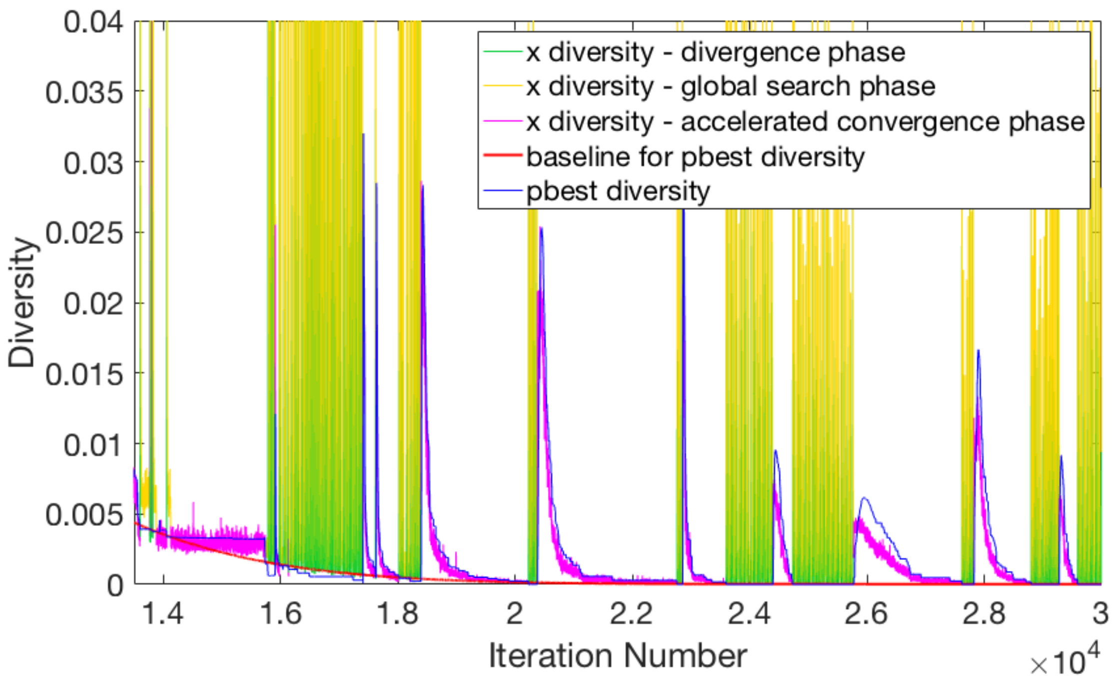


(b)

Fig.1 (a) diversity changes of F_{11} in one run; (b) the second half of Fig.1(a) enlarged



(a)



(b)

Fig.2 (a) diversity changes of F_{14} in one run; (b) the second half of Fig.2(a) enlarged

As Fig.1(a) and Fig.2(a) illustrate, at the beginning of the search process, both the *x diversity* and the *pbest diversity* have a sharp drop. This is because the particles search more globally, and improvement amplitude of the fitness value of the *gbest* position are large. But afterwards, the improvement amplitude would experience a decrease so that the particles converge fast to the neighborhood of the *gbest* position to search more locally. When the value of *x diversity* is very close to that of *pbest diversity*, the *x diversity* strongly fluctuates due to the fast switching between the divergence phase and the global search phase, thus making the particles search more globally. In the later stage of the search process (see Fig.1(b) and Fig.2(b)), sometimes the *pbest diversity* experience a sudden increase, and then quickly decline due to the effectiveness of the accelerated convergence phase. Since in this stage the *pbest* positions of particles are always concentrated in a small area, the sudden increase of the *pbest diversity* demonstrates that a *gbest* position far away from this area has been found, verifying that the DCG strategy can indeed help the algorithm escape from local optima. Another noteworthy phenomenon in the later stage of the search process is that the *pbest diversity* sometimes remains unchanged during a relatively long iterative search process (i.e. the *pbest diversity* between 15500 to 27000 iterations in Fig. 1(b) and the *pbest diversity* between 14000 to 16000 iterations in Fig. 2(b)). During this stage of the search process, the adoption of the accelerated convergence phase makes the particles search more locally to make the *pbest diversity* decline. Once the *pbest diversity* reach the baseline, the rapid fluctuations of the *x diversity* can help the particles search globally temporarily, resulting in that the algorithm has a greater chance to escape from local optima probably.

In addition to the aforementioned similarities, the most significant difference between these two figures is that the curve of *pbest diversity* in Fig.1(a) matches the baseline well, while in some parts of Fig.2(a) (between 1500 to 14000 iterations), the *pbest diversity* decreases much slower than the baseline. The reason may be that the points with better fitness values than all the *pbest* positions of the particles except the *gbest* particle are located in a small neighborhood area around the current *gbest* position. Therefore, when the *pbest diversity* and the *x diversity* are both high, the particles probably search in a relatively large area with a high value of β, the particles have low chance to find this area so that their *pbest* positions are difficult to update. However, the value of β cannot be fixed at a low value (e.g. $\beta = 1.05$), since it can make all the particles quickly drift to the current *gbest* and probably fall into the small neighborhood of a local optima.

Actually, the phenomenon observed in F_{14} (shown in Fig.2) also occurs in some other tested functions. Table 4 illustrates the average percentage of improvements of the *gbest* position made in each phase by the DCG-RDPSO over 51 runs for each benchmark function. It can be observed that, in F_6, F_9, F_{12}, F_{13}, F_{14}, and F_{17}, more than 70% of the gbest improvements are made in the accelerated convergence phase. This means that in these functions, the DCG-RDPSO stays in the accelerated convergence phase in most of the search process. Moreover, the last line in Table 4 further illustrates that the average percentage of the accelerated convergence phase to the whole search process over all tested functions are more than 50%. Note that there are also about 20% of the *gbest* improvements achieved in the divergence phase. Most of these improvements are made in the early stage of the search process, since the value of α at that time should be close to 0.9. Thus, the search behavior of the DCG-RDPSO is similar to that of the CRDPSO in the early stage.

Table 4. The average percentage of the *gbest* improvement made in each phase by the DCG-RDPSO among 51 runs

	Gbest improvement in Divergence phase	Global search phase	Accelerated Convergence phase
F_6	3.03%	17.80%	79.16%
F_7	10.47%	35.49%	54.04%
F_8	62.50%	0.16%	37.34%
F_9	6.10%	14.82%	79.07%
F_{10}	4.63%	83.11%	12.66%
F_{11}	4.44%	44.73%	50.83%
F_{12}	11.80%	9.85%	78.35%
F_{13}	13.44%	11.36%	75.20%
F_{14}	2.10%	19.93%	77.97%
F_{15}	43.32%	0.10%	56.58%
F_{16}	60.78%	0.00%	39.22%
F_{17}	1.92%	16.50%	81.58%
F_{18}	24.51%	42.48%	33.01%
F_{19}	2.83%	29.68%	67.49%
F_{20}	47.42%	5.21%	47.37%
Avg	19.95%	22.08%	57.99%

However, although the accelerated convergence phase plays an important role in the DCG strategy, it does not mean that the RDPSO with only a linear decreasing β can outperform the DCG-RDPSO. To illustrate this clearly, additional experiments were carried out on the RDPSO-Dbeta, for performance comparison, and the results is also shown in Table 3. As can be seen, the results for the RDPSO-Dbeta are almost equivalent to those of the CRDPSO, but worse than those of the DCG-RDPSO for most tested functions. The reason may be that the linearly decreasing β drives the particles to quickly drift to the areas near the current *gbest* position and thus brings about premature convergence. It should be pointed out that if the particles are lucky enough to find the “right area”, they can probably find better results than the CRDPSO and the DCG-RDPSO. This is why on some tested functions, the RDPSO-Dbeta outperforms the other two algorithms for the best final fitness values. Nevertheless, the comparative analysis between RDPSO-Dbeta and DCG-RDPSO shows that the divergence and the global search phases are necessary, and the accelerated convergence phase should be implemented at the right time, so that the algorithm can find a good solution with a high chance to escape the local optimal solution.

4.4 Performance of the DCG-RDPSO for multimodal functions in CEC-2013

In order to further verify the performance of the DCG-RDPSO algorithm, six variants of PSO and four non-PSO evolutionary algorithms were tested on the multimodal functions in CEC-2013, and some statistical results are recorded in Table 5. It should be emphasized here that in this section, the DCG-RDPSO algorithm is not compared with other similar diversity-guided strategies (Riget and Vesterstrøm 2002; Hu et al. 2007; Pant et al. 2007), since we found that the values of diversity in many cases failed to reach the given lower bound when these strategies were tested on the above benchmark functions. For an example, when the attractive and repulsive PSO (ARPSO) (Riget and Vesterstrøm 2002) was tested with the experimental setting mentioned in section 4.1, it was found that the diversity was not able to get to the default lower bound in (Riget and Vesterstrøm 2002) (5.0×10^{-6}) for most multimodal functions, and, as a consequence, the algorithm was always in the “attraction phase”.

Table 5. Statistics on the results of DCG-RDPSO and ten compared algorithms

Function		CPSO	FIPS	CLPSO	CCPSO-ISM	SRPSO	LIPSO	DFnABC	ESGHS	MGDE	SARGA	DCG-RDPSO
F_6	Mean	7.93E+01	4.10E+01	2.46E+01	**1.63E+01**	6.47E+01	1.59E+02	**1.14E+01**	**1.15E+01**	2.90E+01	3.91E+01	2.56E+01
	(t-value)	(11.105)	(6.197)	(5.166)	**(1.787)**	(9.632)	(37.590)	**(0.000)**	**(0.027)**	(4.469)	(6.784)	(5.965)
	Best	2.36E-01	9.75E-02	4.72E+00	1.47E-01	2.22E-01	9.77E+01	8.46E-02	**4.30E-03**	2.76E-01	9.94E+00	1.46E+01
F_7	Mean	3.80E+01	4.54E+01	4.34E+01	6.16E+01	2.35E+01	7.81E+01	6.83E+01	1.58E+02	2.30E+01	1.69E+02	**2.96E+00**
	(t-value)	(21.153)	(32.498)	(42.602)	(44.854)	(15.421)	(26.928)	(49.884)	(19.663)	(12.360)	(24.746)	**(0.000)**
	Best	1.80E+01	2.41E+01	3.14E+01	4.20E+01	6.90E+00	3.85E+01	4.65E+01	6.31E+01	5.35E+00	8.41E+01	**1.96E-01**
F_8	Mean	**2.08E+01**	**2.09E+01**	2.09E+01	**2.08E+01**	2.09E+01	2.09E+01	2.09E+01	2.09E+01	**2.08E+01**	2.09E+01	**2.08E+01**
	(t-value)	**(0.333)**	**(1.511)**	(2.417)	**(0.000)**	(2.369)	(2.042)	(2.209)	(2.353)	**(1.198)**	(4.231)	**(0.984)**
	Best	2.07E+01	2.07E+01	2.07E+01	2.07E+01	2.08E+01	2.07E+01	2.07E+01	2.08E+01	2.07E+01	**2.06E+01**	2.07E+01
F_9	Mean	2.32E+01	1.88E+01	2.24E+01	2.50E+01	2.63E+01	1.85E+01	2.63E+01	2.90E+01	1.97E+01	3.20E+01	**1.08E+01**
	(t-value)	(16.867)	(14.489)	(23.258)	(28.590)	(22.052)	(9.769)	(30.906)	(29.463)	(13.531)	(35.494)	**(0.000)**
	Best	1.17E+01	1.22E+01	1.79E+01	2.10E+01	1.94E+01	9.71E+00	2.06E+01	2.20E+01	1.12E+01	2.66E+01	**4.17E+00**
F_{10}	Mean	1.75E-01	1.02E-01	8.68E-02	1.34E-01	1.15E-01	3.45E+01	2.11E-02	**9.14E-03**	1.38E-01	3.94E-01	4.47E-02
	(t-value)	(13.579)	(15.208)	(29.508)	(26.545)	(13.972)	(8.704)	(5.892)	**(0.000)**	(9.210)	(19.541)	(9.744)
	Best	1.97E-02	2.96E-02	5.34E-02	7.26E-02	2.22E-02	2.28E+00	1.36E-04	**0.00E+00**	1.97E-02	1.46E-01	7.40E-03
F_{11}	Mean	1.01E+01	1.47E+01	**0.00E+00**	5.35E-14	1.21E+01	5.24E+01	3.79E-14	7.74E+01	2.75E+01	8.79E-05	1.02E+00
	(t-value)	(21.378)	(26.753)	**(0.000)**	(28.284)	(23.002)	(25.500)	(10.000)	(30.017)	(23.063)	(5.926)	(3.671)
	Best	3.98E+00	8.49E+00	**0.00E+00**	**0.00E+00**	5.97E+00	2.02E+01	**0.00E+00**	4.08E+01	1.19E+01	2.42E-06	2.76E-10
F_{12}	Mean	6.87E+01	1.02E+02	6.75E+01	1.32E+02	5.46E+01	3.65E+01	9.80E+01	6.98E+02	5.02E+01	3.04E+02	**3.08E+01**
	(t-value)	(12.087)	(23.574)	(17.518)	(26.680)	(9.095)	(2.339)	(20.350)	(24.957)	(7.941)	(26.425)	**(0.000)**
	Best	3.18E+01	6.33E+01	4.69E+01	8.56E+01	3.08E+01	1.53E+01	4.91E+01	2.99E+02	3.20E+01	1.85E+02	**1.20E+01**
F_{13}	Mean	1.26E+02	1.27E+02	1.05E+02	1.85E+02	1.10E+02	1.08E+02	1.44E+02	6.25E+02	7.79E+01	3.47E+02	**6.31E+01**
	(t-value)	(14.288)	(14.923)	(10.585)	(29.561)	(9.054)	(7.456)	(18.148)	(30.147)	(3.159)	(36.778)	**(0.000)**
	Best	7.66E+01	7.12E+01	6.06E+01	1.47E+02	5.73E+01	3.15E+01	1.02E+02	3.95E+02	2.99E+01	2.35E+02	**1.50E+01**
F_{14}	Mean	4.09E+02	2.29E+03	3.14E+00	2.19E+01	3.84E+02	3.55E+03	7.92E-01	1.60E+03	3.36E+02	**4.38E-03**	3.84E+02
	(t-value)	(17.706)	(35.264)	(16.905)	(15.781)	(16.658)	(41.423)	(3.764)	(35.082)	(14.247)	**(0.000)**	(11.277)
	Best	2.75E+01	1.34E+03	5.91E-01	6.89E+00	8.11E+01	2.23E+03	9.20E-02	9.14E+02	3.40E+01	**3.51E-05**	2.52E+01
F_{15}	Mean	6.17E+03	6.13E+03	3.87E+03	**3.07E+03**	6.71E+03	3.28E+03	3.20E+03	4.39E+03	3.26E+03	4.40E+03	4.24E+03
	(t-value)	(28.448)	(52.804)	(12.441)	**(0.000)**	(63.454)	(2.164)	(2.200)	(13.224)	(2.204)	(11.454)	(6.514)
	Best	3.69E+03	5.39E+03	2.62E+03	2.40E+03	5.97E+03	1.84E+03	2.56E+03	3.02E+03	2.17E+03	2.86E+03	**1.71E+03**
F_{16}	Mean	1.85E+00	1.96E+00	1.86E+00	7.09E-01	2.01E+00	1.86E+00	5.47E-01	**4.79E-02**	1.21E+00	1.53E+00	1.97E+00
	(t-value)	(49.470)	(57.901)	(58.398)	(27.951)	(71.851)	(26.670)	(27.638)	**(0.000)**	(33.254)	(28.835)	(59.837)
	Best	1.09E+00	1.39E+00	1.37E+00	4.26E-01	1.36E+00	8.28E-02	1.98E-01	**1.53E-02**	6.35E-01	7.04E-01	1.05E+00
F_{17}	Mean	4.65E+01	1.21E+02	**3.05E+01**	3.24E+01	4.52E+01	5.66E+01	**3.05E+01**	4.30E+01	4.82E+01	**2.98E+01**	3.50E+01
	(t-value)	(17.499)	(66.168)	**(1.128)**	(4.175)	(16.478)	(24.700)	**(1.015)**	(6.545)	(17.658)	**(0.000)**	(5.805)
	Best	3.71E+01	1.05E+02	3.05E+01	3.15E+01	3.56E+01	4.44E+01	3.04E+01	3.07E+01	3.42E+01	**4.21E-03**	3.06E+01
F_{18}	Mean	1.74E+02	1.83E+02	1.46E+02	1.32E+02	1.67E+02	**6.51E+01**	9.18E+01	2.19E+02	9.45E+01	3.88E+02	1.35E+02
	(t-value)	(16.635)	(62.876)	(36.039)	(26.401)	(14.394)	**(0.000)**	(11.005)	(27.127)	(10.933)	(23.491)	(14.201)
	Best	6.35E+01	1.60E+02	1.07E+02	1.02E+02	7.03E+01	**4.89E+01**	6.85E+01	1.37E+02	6.34E+01	1.79E+02	5.91E+01
F_{19}	Mean	2.55E+00	7.81E+00	1.67E+00	6.37E-01	2.22E+00	9.76E+00	**3.18E-02**	3.83E+00	2.48E+00	9.15E-01	1.50E+00
	(t-value)	(25.470)	(31.173)	(72.593)	(16.024)	(25.081)	(23.837)	**(0.000)**	(27.099)	(25.176)	(19.334)	(23.189)
	Best	7.03E-01	5.03E+00	1.28E+00	2.62E-01	1.05E+00	3.82E+00	**5.68E-14**	1.79E+00	6.06E-01	3.00E-01	1.01E+00
F_{20}	Mean	1.10E+01	1.18E+01	1.12E+01	1.23E+01	1.05E+01	1.47E+01	1.11E+01	1.43E+01	1.01E+01	1.40E+01	**9.68E+00**
	(t-value)	(7.235)	(12.604)	(12.199)	(17.756)	(4.562)	(31.857)	(5.201)	(30.484)	(2.471)	(25.876)	**(0.000)**
	Best	8.94E+00	1.07E+01	1.05E+01	1.09E+01	8.31E+00	1.13E+01	9.19E+00	1.10E+01	8.35E+00	1.15E+01	**7.75E+00**
Best-Mean[1]		1	1	2	3	0	1	3	3	1	2	**6**
Best-Best[2]		0	0	1	1	0	1	2	3	0	3	**6**
Best-Both[3]		0	0	1	0	0	1	1	3	0	2	**5**

[1] The number of mean results that are bold of the corresponding algorithm
[2] The number of best results that are bold of the corresponding algorithm
[3] The number of functions in which the corresponding algorithm can get the best or equivalently best results in both mean values and best values at the same time

In Table 5, with respect to the mean and best of the final fitness values for each function, the best result among all the algorithms and the equivalently best ones are also in bold. According to the mean values listed in the table, the proposed DCG-RDPSO algorithm can get the best results among all the compared algorithms for the most tested functions, followed by DFnABC, ESGHS, CCPSO-ISM, and then CLPSO and SARGA, while the other algorithms can only obtain at most one smallest mean value. Moreover, the t-values in Table 5 further illustrate that in almost every function (except F_6, F_8 and F_{17}), the superiority of the algorithm that can obtain the best mean value is statistically significant. Put it in detail, excluding F_8, the proposed algorithm outperforms other algorithms for F_7, F_9, F_{12}, F_{13} and F_{20}. Moreover, as Table 6 shows, for the functions that the DCG-RDPSO cannot get the smallest mean values, the proposed algorithm is ranked second to fourth among eleven competitors in almost each of these functions only except F_{16}. This shows the good robustness of the DCG-RDPSO in solving most multimodal problems.

Table 6. The rank of the mean values and best values of DCG-RDPSO in each function among all tested algorithms

Function	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}
Rank of Mean[1]	4	1	3	1	3	2	1	1
Rank of Best	10	1	3	1	3	4	1	1
Function	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}	
Rank of Mean	3	4	10	4	4	3	1	
Rank of Best	4	1	7	4	2	6	1	

[1] Ranked according to the t-values in Table 5

As for the best values of the final fitness in Table 5, the DCG-RDPSO can also obtain the best results among all the compared algorithms on six test functions. For most of the other functions, the difference between the best value yielded by the proposed algorithm and the smallest one among all the algorithms is not statistically significant. Furthermore, as shown in Table 6, the best values of the proposed algorithm are ranked top four for all the tested functions except F_6, F_{16} and F_{19}. It means that the proposed algorithm has good robustness in finding solutions with higher quality, again demonstrating that the accelerated convergence operators in the DCG strategy can definitely enhance the search ability of the RDPSO. It should be pointed out that for those tested functions, the mean values of which the DCG-RDPSO obtained are the smallest, the algorithm can also get the smallest best values at the same time (see the Best-Both in Table 5). This is very important in solving some real-world problems, e.g. in flexible ligand docking (Chen et al. 2007; Morris et al. 1998), and in ED problem (Chowdhury and Rahman 1990; Chokpanyasuwan 2009; see statistical results in Tables 8 to 10). In comparison, although the CCPSO-ISM algorithm can achieve the smallest mean values for three functions, it cannot obtain the smallest best value for any of these functions.

Additionally, the mean computational time taken by all the compared PSO algorithms was evaluated to show the influence of the DCG strategy on the computational complexity. The results in Table 7 illustrate that, compared with CRDPSO, RDPSO-Dbeta and CPSO, the mean computational time consumed by the proposed algorithm is longer, since calculating the diversity values in each iteration inevitably takes the proposed algorithm extra time. However, the DCG-RDPSO is less time-consuming than all the other algorithms, verifying that the DCG strategy is not as complicated as the strategies used in other compared PSO variants. It should be noted that according to Table 7, the difference between the mean computational time of DCG-RDPSO and those of CRDPSO, RDPSO-Dbeta and CPSO is relatively small, because the time spent in calculating the diversity values is generally much less than that in calculating the object function values.

Table 7. Mean computational time of a single run taken by all the compared PSO algorithms for testing all the sixteen multimodal benchmark functions

Algorithm	DCG-RDPSO	CRDPSO	RDPSO-Dbeta	CPSO	FIPS
Mean time	54.628s	51.366s	51.438s	52.559s	65.449s
Algorithm	CLPSO	CCPSO-ISM	SRPSO	LIPSO	
Mean time	56.710s	58.775s	61.816s	150.225s	

In conclusion, as can be observed from Tables 5 to 7, the DCG-RDPSO algorithm is able to get the best (or equivalently best) results in both mean values and best values for most of the tested functions among all the compared algorithms, and its performance robustness and computational complexity are also impressive.

4.5 Performance of the DCG-RDPSO for ED problems

Economic dispatch is one of the most important problems in operation of power systems. When the production schedule for a period of time of a power station should be prepared in advance, this issue needs to be considered. Its purpose is to minimize the total short-term costs of operating the generators by appropriately dividing the total load demand among available generators to meet the predicted customer load (Chowdhury and Rahman 1990). The objective function of an ED problem can be formulated as the following:

$$\text{Minimize} \qquad F_{cost} = \sum_{j=1}^{N_g} F_j(P_j) \tag{23}$$

where P_j is the real output of the jth generating units (in MW), N_g is the total number of generators in the power system, and $F_j(P_j)$ is the cost function of the jth generating unit (in \$/h), and is typically modeled by a smooth quadratic function

$$F_j(P_j) = a_j + b_j P_j + c_j P_j^2 \tag{24}$$

where a_j, b_j and c_j are the cost coefficients of the jth generating unit.

However, the generator sets with multi-valve steam turbines always show greater changes in the cost function. Considering the valve point effects, the cost function should contain higher-order nonlinearities, which can be expressed as:

$$F_j(P_j) = a_j + b_j P_j + c_j P_j^2 + |e_j \times \sin(f_j \times (P_j^{min} - P_j))| \tag{25}$$

where e_j and f_j are the cost coefficients of the jth generating unit reflecting the valve-point effects (Sinha et al. 2003), and P_j^{min} is the lower production limit of the jth unit.

The ED problem should satisfy various nonlinear constraints, i.e. power balance, generation limits, ramp rate limits, and prohibited operating zones (Swarup and Yamashiro 2002). All these constraints are expressed in detail below (Zhao et al. 2018).

A. Power balance

The power balance constraint restricts that the total system generation should meet the power demand and transmission losses

$$\sum_{j=1}^{N_g} P_j = P_T + P_L \tag{26}$$

where P_T is the total power demand (in MW), and P_L, which is the transmission network loss (in MW), is generally computed by using B-coefficients (Wood and

Wollenberg 2003) and is given by

$$P_L = \sum_{j=1}^{N_g}\sum_{k=1}^{N_g} P_j B_{jk} P_k + \sum_{j=1}^{N_g} P_j B_{j0} + B_{00} \tag{27}$$

where B_{jk}, B_{j0} and B_{00} are known as the loss coefficients. $[B_{jk}]$ is an $N_g \times N_g$ matrix, in which the elements are all constants under normal conditions.

B. Generation Limits

The power generation of each unit should vary between its lower production limit (P_j^{min}) and upper one (P_j^{max}):

$$P_j^{min} \leq P_j \leq P_j^{max}, (j = 1, 2, \cdots, N_g) \tag{28}$$

C. Ramp Rate Limits

Practically, the operating range of all online units should be restricted by their ramp rate limits. The increase and reduction of power generation in each generator are limited by

$$P_j - P_j^0 \leq UR_j \tag{29}$$

$$P_j^0 - P_j \leq DR_j \tag{30}$$

where P_j^0 (in MW) is the previous output power, UR_j (in MW/h) and DR_j (in MW/h) are the up-ramp and down-ramp limit of the jth generator. Combining (29) and (30) with (28) results in

$$\max(P_j^{min}, P_j^0 - DR_j) \leq P_j \leq \min(P_j^{max}, P_j^0 + UR_j) \tag{31}$$

D. Prohibited Operating Zones

In actual power system, opening or closing the steam valve always causes some prohibited operating zones. Considering the constraint in (28), and to make the load demand of a power system avoid the prohibited zones, the feasible operating zones of the jth generator can be described as follows:

$$P_j \in \begin{cases} P_j^{min} \leq P_j \leq P_{j,1}^{l} \\ P_{j,k-1}^{u} \leq P_j \leq P_{j,k}^{l}\ , k = 2, 3, \cdots, n_j \\ P_{j,n_j}^{u} \leq P_j \leq P_j^{max} \end{cases} \tag{32}$$

where n_j is the number of prohibited operating zones of the jth generator, and $P_{j,k}^{u}$ and $P_{j,k}^{u}$ are the lower and upper bounds of power in the kth prohibited zones by the jth generator, respectively.

All these four constraints and the objective function together describe the ED problem mathematically. Since the ED is a multimodal problem, which has nonsmooth and nonconvex cost function along with nonlinear constraints, particle swarm algorithms and some other evolutionary algorithms are effective methods to address this issue. In this paper, all the aforementioned algorithms were examined on three practical power systems. System 1 consists of 15 thermal units with prohibited operating zones, ramp rate limits, and transmission network loss. The load demand of system 1 is 2630MW. The units' characteristics and the loss coefficients B of system 1 can be accessed in (Alomoush and Oweis 2018). System 2 consists of 40 generating units with valve-point effects. It is adapted from (Chen and Chang 1995), with modifications to incorporate the valve-point loading, and its load demand is set to 10500MW. The input data of system 2 are described in (Sinha et al. 2003). System 3 is a 140-unit Korean power system with valve-point effects, prohibited operating zones, and ramp rate limits. The load demand is set to 49342 MW and the input data are given in (Park et al. 2009).

When evolutionary algorithms are applied to the ED problems, a key issue is how to handle constraints with efficiency. In this paper, the power balance constraint is addressed by adding a penalty term to (23) (Meng et al. 2015). Thus, the objective function can be rewritten as

$$\text{Minimize} \qquad F_{cost} = \sum_{j=1}^{N_g} F_j(P_j) + K\left|\sum_{j=1}^{N_g} P_j - P_D - P_L\right| \tag{33}$$

where K is the penalty coefficient. If the power balance constraint is violated when solving ED problems, the penalty term works by adding a large number to the objective function given by equation (23). Therefore, the candidate solutions violating (26) should have a relatively large objective function value and will be eliminated gradually during the search process. In this paper, K was set to 100 for all the three tested power systems.

Before evaluate the candidate solution by using equation (33), any variable in the solution cannot be in the range of the prohibited operation area. Therefore, a solution containing one or more variables within the prohibited operation area is penalized with a very large positive constant, to ensure this kind of solution will be abandoned.

For any variable of a solution that violates the generation limits or the ramp-rate limits, our constraint handling strategy is applied to make P_j move either towards the lower bound or the upper bound of its value range according to equation (34).

$$P_j = \begin{cases} \min(P_j^{max}, P_j^0 + UR_j), & \text{if } P_j > \min(P_j^{max}, P_j^0 + UR_j) \\ \max(P_j^{min}, P_j^0 - DR_j), & \text{if } P_j < \max(P_j^{min}, P_j^0 - DR_j) \\ P_j, & \text{otherwise} \end{cases} \tag{34}$$

Tables 8 to 10 illustrate the statistical results of the total costs and the computational time obtained by each method for the ED problem of system 1 to 3, respectively. The t-values listed in these three tables are measured by comparing the mean costs of the DCG-RDPSO and those of the other compared algorithms. According to these tables, it is clear that the DCG-RDPSO algorithm can get the best results in almost every cost-related evaluation criterion, only except the minimum cost for system 1 and the standard deviation for system 3. More specifically, for all of the three power systems, the mean cost and maximum cost obtained by the variants of RDPSO (i.e. CRDPSO, RDPSO-Dbeta, and DCG-RDPSO) are much better than those of the other algorithms. With respect to the variants of RDPSO, although their mean costs are comparable to each other for the 15-unit system (the t-value obtained by CRDPSO and RDPSO-Dbeta are both smaller than 1.984 for system 1), the advantage of DCG-RDPSO over CRDPSO and RDPSO-Dbeta is obvious for the systems with higher dimensionalities, demonstrating the effectiveness of the DCG strategy. Furthermore, the standard deviation of DCG-RDPSO is also better than those of CRDPSO and RDPSO-Dbeta in each tested power system. This indicates that the proposed DCG strategy can definitely enhance the robustness of the proposed algorithm. As the dimensionality of the problems increases, the superiority of the DCG-RDPSO is also reflected in the minimum cost. For system 1, the minimum cost found by DCG-RDPSO is very close to those found by several other algorithms (i.e. CRDPSO, RDPSO-Dbeta, CPSO, SRPSO and MGDE), but for the other two systems, the DCG-RDPSO can find much smaller minimum costs than all the other algorithms.

It should be pointed out that although CCPSO-ISM, DFnABC, ESGHS and SARGA can get better results than DCG-RDPSO for some CEC-2013 benchmark functions according to Table 5, the cost-related statistical results of the ED problems in this section indicate that the DCG-RDPSO is superior to these algorithms, especially for the high-dimensional systems (system 2 and system 3). According to Tables 8 to 10, another advantage of the DCG-RDPSO over DFnABC and ESGHS is that the DCG-RDPSO takes much less computational time than these two algorithms when solving ED problems. The reason is that in each iteration, the DFnABC algorithm needs to traverse a certain dimension of all food source positions multiple times to calculate various kinds of distances (Cui et al. 2018), and the ESGHS should find the worst solution among all the solutions in the harmony memory (Luo et al. 2019), which significantly reduces the efficiency of these two algorithms.

Table 8. Algorithm comparison for System 1

	Min. Cost	Mean Cost	Std. Dev.	Max. Cost	T-value	Mean Time
CRDPSO	32645.9011	32648.8148	17.4635	32771.0171	0.8501	3.3884s
RDPSO-Dbeta	32645.8877	32650.0487	12.2873	32699.5838	1.9224	3.4646s
CPSO	32645.8626	32708.3769	79.0026	32977.1075	5.5719	3.6999s
FIPS	32710.7139	32784.9386	33.6305	32867.0956	29.3387	5.7585s
CLPSO	32646.0123	32663.4135	16.6970	32727.0185	7.1252	3.6964s
CCPSO-ISM	32753.0063	32851.3044	43.6347	32951.5239	33.4747	4.1976s
SRPSO	32645.8620	32654.5579	22.2930	32742.6581	2.5047	5.7371s
LIPSO	32932.8640	33104.9754	88.9523	33268.0864	36.7878	24.3035s
DFnABC	32844.7597	32927.6364	35.9971	33008.0859	55.7129	17.1635s
ESGHS	32657.5012	32781.9048	90.2803	33097.2414	10.6920	67.7188s
MGDE	**32645.8617**	32657.6435	23.2546	32744.7960	3.3483	4.0841s
SARGA	32838.7315	33019.2272	83.1136	33223.8652	32.0044	5.4489s
DCG-RDPSO	32645.8793	**32646.7335**	**0.83905**	**32648.8076**		3.4828s

Table 9. Algorithm comparison for System 2

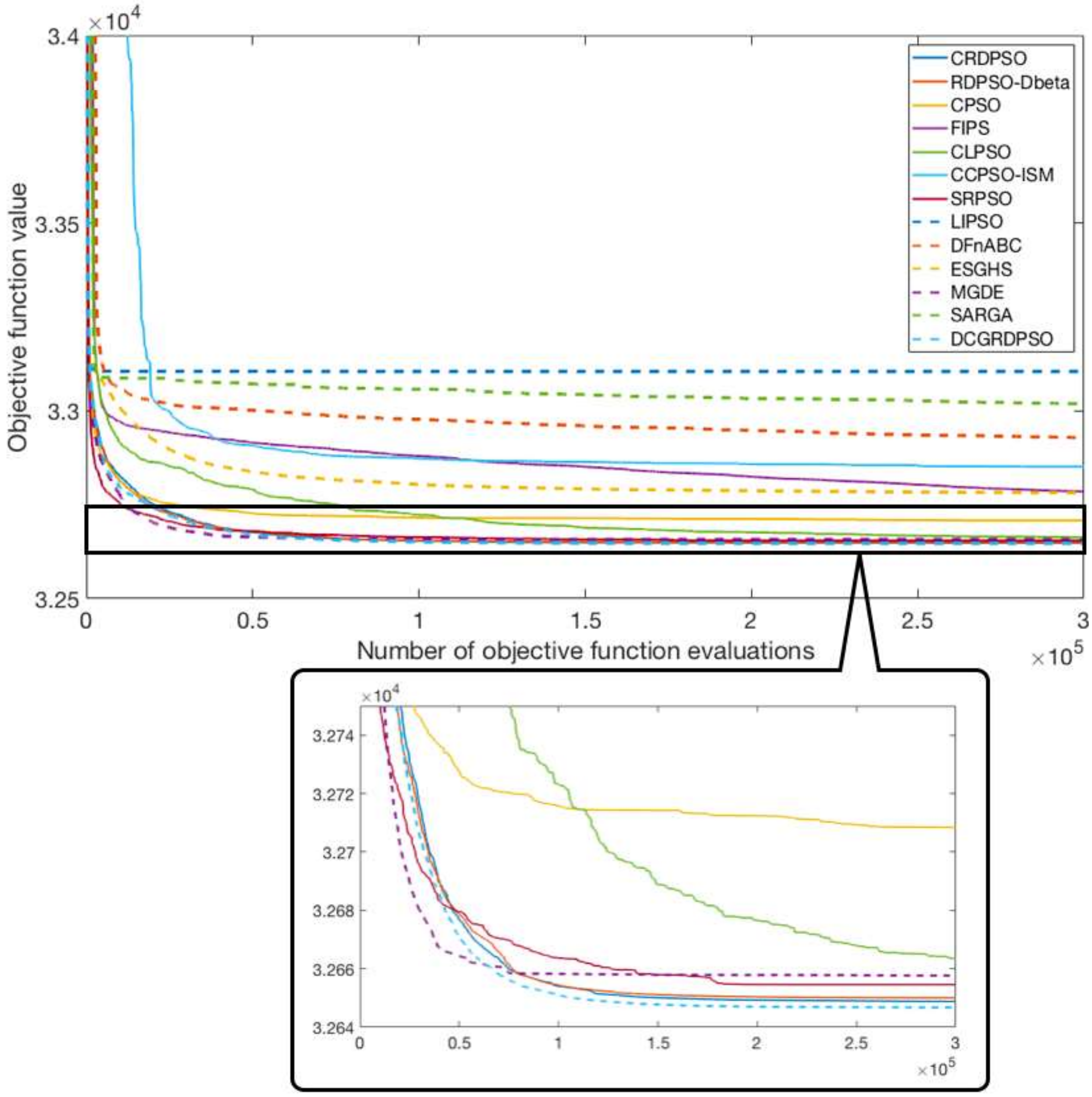
	Min. Cost	Mean Cost	Std. Dev.	Max. Cost	T-value	Mean Time
CRDPSO	121540.775	121974.637	399.291	122972.256	2.477	2.0850s

RDPSO-Dbeta	121525.651	122028.599	316.171	122832.936	4.065	1.9651s
CPSO	124522.960	127449.206	1935.292	137467.413	20.677	1.9391s
FIPS	122125.854	122709.112	271.746	123585.648	19.681	6.0584s
CLPSO	123079.831	123860.384	269.944	124372.510	45.495	2.1028s
CCPSO-ISM	125643.100	126562.806	352.625	127167.216	86.334	2.4459s
SRPSO	123544.417	124566.112	761.101	126374.593	25.102	4.8085s
LIPSO	128364.270	132352.906	1756.686	136786.490	42.601	26.5188s
DFnABC	130915.014	134340.868	1770.928	138607.723	50.241	15.0021s
ESGHS	122837.159	125737.424	885.068	127621.951	31.002	166.8165s
MGDE	121632.627	121980.113	216.394	122603.362	4.042	2.9415s
SARGA	124196.816	125924.545	830.826	128393.699	34.522	5.0409s
DCG-RDPSO	**121481.842**	**121823.957**	**171.212**	**122272.178**		2.1329s

Table 10. Algorithm comparison for System 3

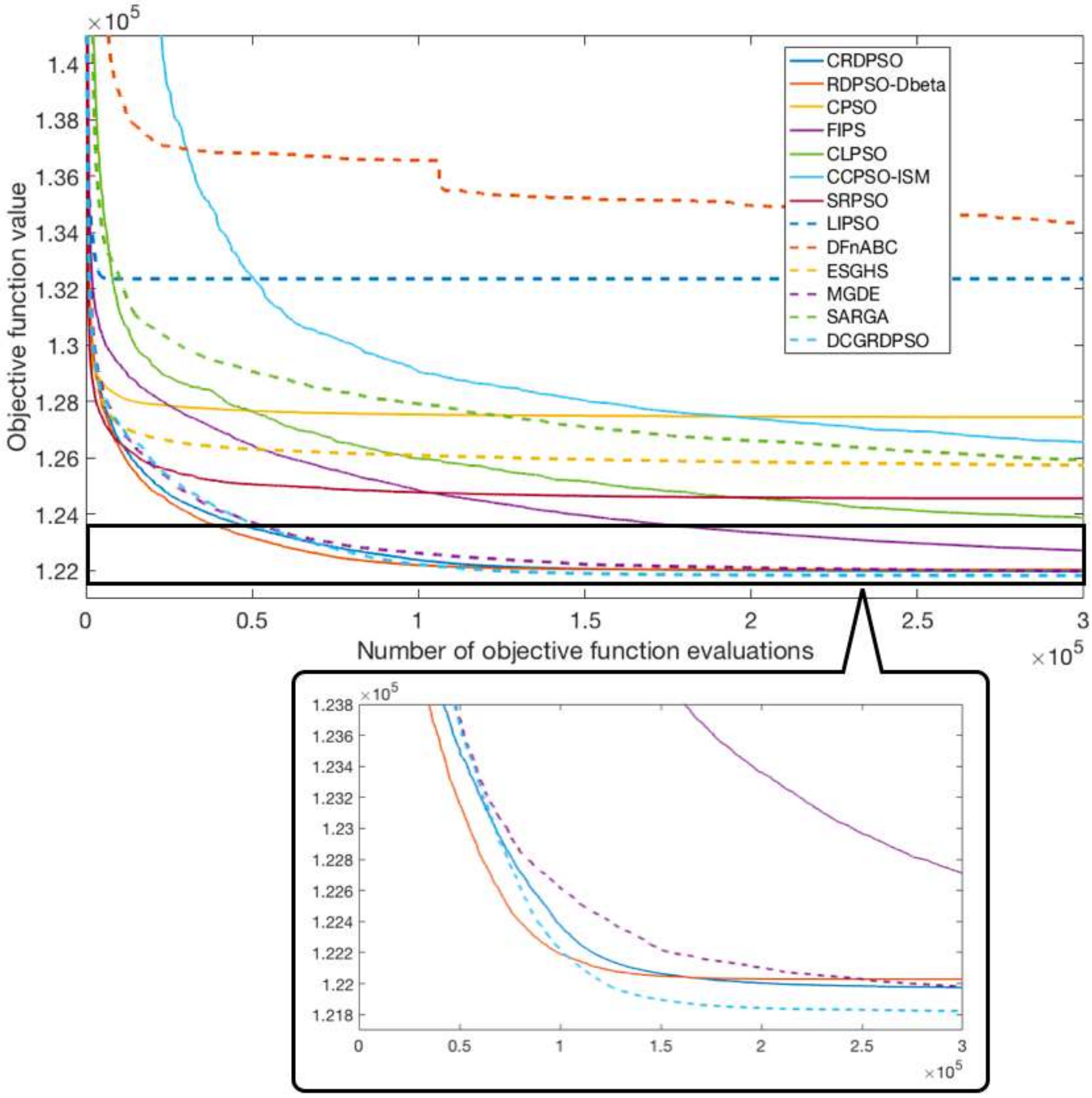
	Min. Cost	Mean Cost	Std. Dev.	Max. Cost	T-value	Mean Time
CRDPSO	1658045.93	1659277.88	1308.58	1662842.71	4.07	15.1219s
RDPSO-Dbeta	1658080.62	1658936.44	801.98	1662163.81	3.42	14.855s
CPSO	1723811.80	1769446.92	23737.26	1821888.23	33.37	15.5844s
FIPS	1660730.35	1661987.43	776.50	1666148.62	28.61	18.1654s
CLPSO	1659238.93	1660230.31	384.28	1661029.77	22.50	16.3159s
CCPSO-ISM	1695759.21	1709045.81	4573.86	1715507.24	78.62	14.9456s
SRPSO	1687971.59	1716153.31	13703.42	1744368.37	30.03	19.6805s
LIPSO	1726638.89	1812050.04	42662.09	1892142.64	25.70	47.7224s
DFnABC	1894682.28	1920252.74	13234.63	1969680.08	141.18	27.9847s
ESGHS	1667101.11	1680913.83	7165.46	1698333.27	22.30	699.6723s
MGDE	1659062.85	1659720.44	**319.09**	1660479.31	17.21	16.4821s
SARGA	1659822.35	1661258.04	863.55	1663871.21	20.73	20.9469s
DCG-RDPSO	**1657996.98**	**1658510.35**	387.82	**1660023.36**		16.5239s

In order to visualize how these compared algorithms are reducing the object function values with respect to the number of function evaluations, we plotted in Fig. 3 the convergence curves of each tested algorithm averaged over 51 runs on the ED problem of each power system. As shown in Fig. 3, the convergence curves of DCG-RDPSO are all somewhat similar to the downward trend of the baseline in Fig. 1 and Fig. 2. The beginning convergence speed of DCG-RDPSO is not too fast (slower than MGDE and SRDPSO in Fig. 3(a), and slower than CRDPSO and RDPSO-Dbeta in Fig. 3(b) and 3(c)), which ensures enough global search behaviors. Moreover, the characteristics of RDPSO itself and the accelerated convergence operations in the DCG strategy together guarantee that the DCG-RDPSO has fast enough convergence speed during the whole search process. In the later stage of the search process, the DCG-RDPSO keeps searching without stagnation and can finally find the best mean result among all compared algorithms for each ED problem. All these phenomena demonstrate that DCG-RDPSO yielded better convergence properties than its competitors. Since DCG-RDPSO has better performance than most of the compared algorithms for both benchmark problems (CEC-2013) and real-world problems (ED), it can be concluded that the proposed algorithm could be a promising optimizer for solving multimodal problems.



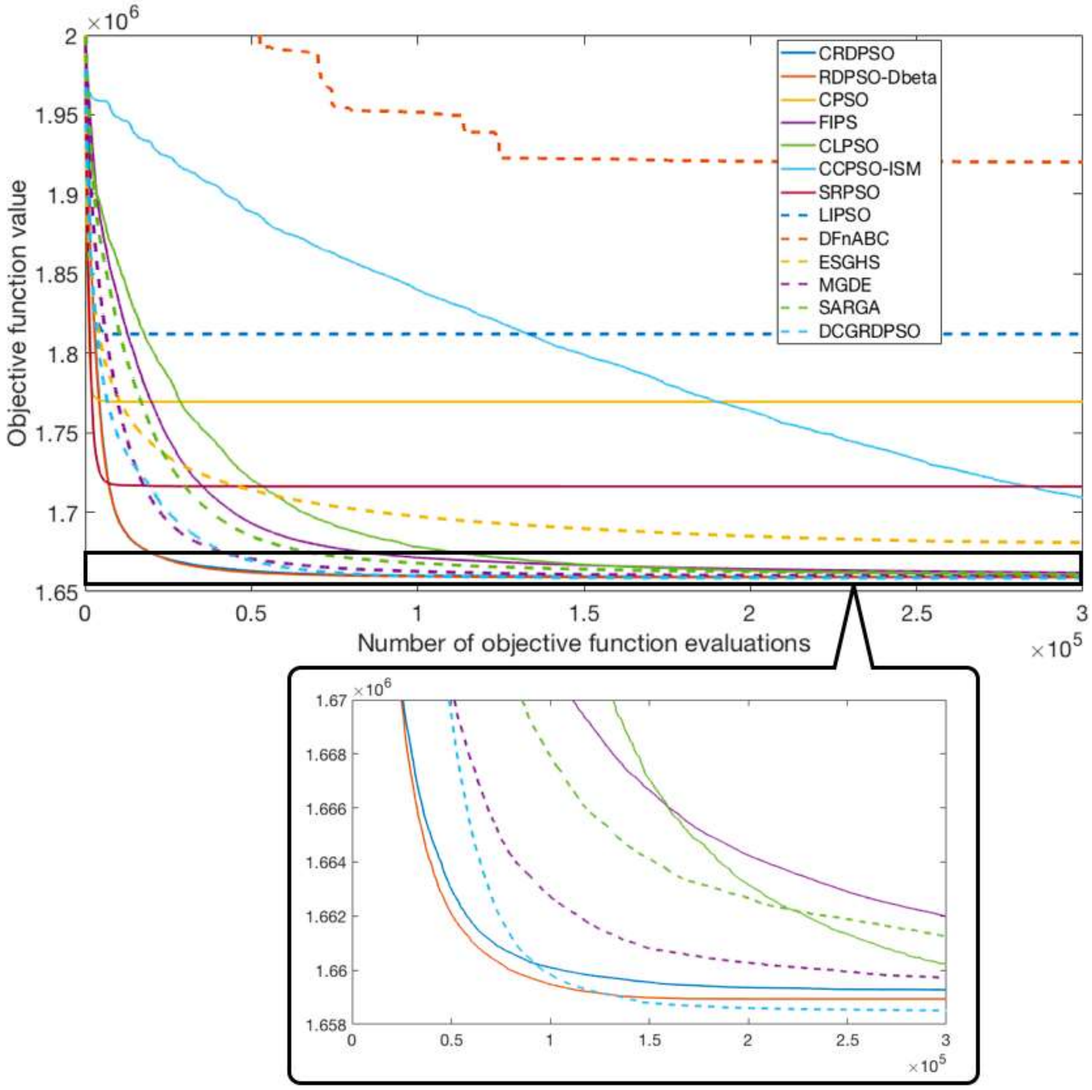
×10^4
3.4
3.35
3.3
3.25
Objective function value
CRDPSO
RDPSO-Dbeta
CPSO
FIPS
CLPSO
CCPSO-ISM
SRPSO
LIPSO
DFnABC
ESGHS
MGDE
SARGA
DCGRDPSO
0
0.5
1
1.5
2
2.5
3
Number of objective function evaluations
×10^5
×10^4
3.274
3.272
3.27
3.268
3.266
3.264
×10^5

(a)



×10^5
1.4
1.38
1.36
1.34
1.32
1.3
1.28
1.26
1.24
1.22
Objective function value
0
0.5
1
1.5
2
2.5
3
Number of objective function evaluations
×10^5
CRDPSO
RDPSO-Dbeta
CPSO
FIPS
CLPSO
CCPSO-ISM
SRPSO
LIPSO
DFnABC
ESGHS
MGDE
SARGA
DCGRDPSO
×10^5
1.238
1.236
1.234
1.232
1.23
1.228
1.226
1.224
1.222
1.22
1.218
0
0.5
1
1.5
2
2.5
3
×10^5

(b)



(c)

Fig.3 Convergence curves of the tested evolutionary algorithms for (a) 15-unit system; (b) 40-unit system; (c) 140-unit system

5. Conclusions

This paper proposed a novel diversity-guided strategy for the RDPSO algorithm that improves the global and local search abilities of the RDPSO algorithm. This strategy divides the RDPSO algorithm into several phases based on a pre-set baseline in order to make the *x diversity* and *pbest diversity* change in a collaborative manner. The operators in the divergence phase and the global search phase promote the algorithm to search more globally, and the proper parameter setting in the accelerated convergence phase drives the particles to search in a smaller area to enhance the local search ability of the algorithm. The experimental results firstly proved the effectiveness of the DCG strategy in improving the overall performance of the RDPSO algorithm, and, secondly, showed that the novel DCG-RDPSO algorithm is a promising algorithm in solving multimodal problems, due to its good performance and high robustness. In future work, we will try to modify this proposed strategy in two ways: one is to set the baseline in an automatic way, and the other is to further generalize the strategy to other PSO variants.

Declarations

Funding

This work was supported in part by the National Natural Science Foundation of China (Projects Numbers: 61673194, 61672263, 61672265), and in part by the national first-class discipline program of Light Industry Technology and Engineering (Project Number: LITE2018-25).

Conflicts of interest/Competing interests

The authors declare that they have no conflict of interest.

Availability of data and material

The datasets analysed during the current study are available in the ref (Liang et al. 2013), (Alomoush and Oweis 2018), (Sinha et al. 2003) and (Park et al. 2009).

Code availability

Pseudocode is available in this article.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Projects Numbers: 61673194, 61672263, 61672265), and in part by the national first-class discipline program of Light Industry Technology and Engineering (Project Number: LITE2018-25).

References

Alomoush MI, Oweis ZB (2018) Environmental - economic dispatch using stochastic fractal search algorithm. Int Trans Electr Energy Syst 28(5):e2530

Assareh E, Behrang MA, Assari MR, Ghanbarzadeh A (2010) Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran. Energy 35(12):5223-5229

Brits R, Engelbrecht AP, Van den Bergh F (2002) A niching particle swarm optimizer. In Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning, Singapore: Orchid Country Club, vol 2, pp 692-696

Cao Y, Zhang H, Li W, Zhou M, Zhang Y, Chaovalitwongse WA (2018) Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions. IEEE Trans Evol Comput 23(4):718-731

Chen PH, Chang HC (1995) Large-scale economic dispatch by genetic algorithm. IEEE Trans Power Syst 10(4):1919-1926

Chen HM, Liu BF, Huang HL, Hwang SF, Ho SY (2007) SODOCK: Swarm optimization for highly flexible protein–ligand docking. J Comput Chem 28(2):612-623

Chokpanyasuwan C (2009) Honey bee colony optimization to solve economic dispatch problem with generator constraints. In 2009 6th International Conference on Electrical

Engineering/Electronics, Computer, Telecommunications and Information Technology, IEEE, vol 1, pp 200-203

Chowdhury BH, Rahman S (1990) A review of recent advances in economic dispatch. IEEE Trans Power Syst 5(4):1248-1259

Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58-73

Cui L, Zhang K, Li G, Wang X, Yang S, Ming Z, Huang JZ, Lu N (2018) A smart artificial bee colony algorithm with distance-fitness-based neighbor search and its application. Future Gener Comput Syst 89:478-493

Du WB, Gao Y, Liu C, Zheng Z, Wang Z (2015) Adequate is better: particle swarm optimization with limited-information. Appl Math Comput 268:832-838

Hu J, Zeng J, Tan Y (2007) A diversity-guided particle swarm optimizer for dynamic environments. In International Conference on Life System Modeling and Simulation, Springer, Berlin, Heidelberg, pp 239-247

Janostik J, Pluhacek M, Senkerik R, Zelinka I (2016) Particle swarm optimizer with diversity measure based on swarm representation in complex network. In Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015, Springer, Cham, pp 561-569

Kao YT, Zahara E (2008) A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl Soft Comput 8(2):849-857

Kennedy J (2006) Swarm intelligence. In: Handbook of nature-inspired and innovative computing. Springer, Boston, MA, pp 187-219

Kennedy J, Eberhart R (1995) Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks, IEEE, vol 4, pp 1942-1948

Kittel C, Kroemer H (1998) Thermal physics (2nd ed.). W.H. Freeman, San Francisco

Li X (2007) A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio. In Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp 78-85

Li X (2009) Niching without niching parameters: particle swarm optimization using a ring topology. IEEE Trans Evol Comput 14(1):150-169

Li Z, Ngambusabongsopa R, Mohammed E, Eustache N (2011) A novel diversity guided particle swarm multi-objective optimization algorithm. Int J Digit Content Technol Appl 5(1):269-278

Li Y, Zhan Z H, Lin S, Zhang J, Luo X (2015) Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. Inf Sci 293:370-382

Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10(3):281-295

Liang JJ, Qu BY, Suganthan PN, Hernández-Díaz, AG (2013) Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, 201212(34):281-295

Liu Q, Du S, van Wyk BJ, Sun Y (2020) Niching particle swarm optimization based on Euclidean distance and hierarchical clustering for multimodal optimization. Nonlinear Dyn 99(3):2459-2477

Liu J, Ma D, Ma TB, Zhang W (2017) Ecosystem particle swarm optimization. Soft Comput 21(7):1667-1691

Luo K, Ma J, Zhao Q (2019) Enhanced self-adaptive global-best harmony search without any extra statistic and external archive. Inf Sci 482:228-247

Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. IEEE Trans Evol Comput 8(3):204-210

Meng A, Hu H, Yin H, Peng X, Guo Z (2015) Crisscross optimization algorithm for large-scale dynamic economic dispatch problem with valve-point effects. Energy 93:2175-2190

Mohammadi F, Abdi H (2018) A modified crow search algorithm (MCSA) for solving economic load dispatch problem. Appl Soft Comput 71:51-65

Morris GM, Goodsell DS, Halliday RS, Huey R, Hart WE, Belew RK, Olson AJ (1998) Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. J Comput Chem 19(14):1639-1662

Omar MA (1975) Elementary solid state physics: principles and applications. Pearson Education India

Pant M, Radha T, Singh V P (2007) A simple diversity guided particle swarm optimization. In 2007 IEEE Congress on Evolutionary Computation, IEEE, pp 3294-3299

Park JB, Jeong YW, Shin JR, Lee KY (2009) An improved particle swarm optimization for nonconvex economic dispatch problems. IEEE Trans Power Syst 25(1):156-166

Riget J, Vesterstrøm JS (2002) A diversity-guided particle swarm optimizer-the ARPSO. Dept Comput Sci, Univ of Aarhus, Aarhus, Denmark, Tech Rep, 2:2002

Ruxton GD (2006) The unequal variance t-test is an underused alternative to Student's t-test and the

Mann–Whitney U test. Behav Ecol 17(4):688-690

Sareni B, Krahenbuhl L (1998) Fitness sharing and niching methods revisited. IEEE Trans Evol Comput 2(3):97-106

Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence, IEEE, pp 69-73

Sinha N, Chakrabarti R, Chattopadhyay PK (2003) Evolutionary programming techniques for economic load dispatch. IEEE Trans Evol Computat 7(1):83–94

Subbaraj P, Rengaraj R, Salivahanan S (2009) Enhancement of combined heat and power economic dispatch using self adaptive real-coded genetic algorithm. Appl Energy 86(6):915-921

Sun J, Palade V, Wu X J, Fang W, Wang Z (2013) Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization. IEEE Trans Industr Inform 10(1):222-232

Sun J, Wu X, Palade V, Fang W, Shi Y (2015) Random drift particle swarm optimization algorithm: convergence analysis and parameter selection. Mach Learn 101(1-3):345-376

Sun J, Zhao J, Wu X, Fang W, Cai Y, Xu W (2010) Parameter estimation for chaotic systems with a drift particle swarm optimization method. Phys Lett A 374(28):2816-2822

Swarup KS, Yamashiro S (2002) Unit commitment solution methodology using genetic algorithm. IEEE Trans Power Syst 17(1):87-91

Tanweer M R, Suresh S, Sundararajan N (2015) Self regulating particle swarm optimization algorithm. Inf Sci 294:182-202

Ursem R K (2002) Diversity-guided evolutionary algorithms. In International Conference on

Parallel Problem Solving from Nature, Springer, Berlin, Heidelberg, pp 462-471

Vitela JE, Castaños O (2008, June) A real-coded niching memetic algorithm for continuous multimodal function optimization. In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), IEEE, pp 2170-2177

Wang H, Sun H, Li C, Rahnamayan S, Pan JS (2013) Diversity enhanced particle swarm optimization with neighborhood search. Inf Sci 223:119-135

Wood AJ and Wollenberg BF (2003) Power Generation, Operation, and Control. Beijing: Tsinghua University Press, pp 195

Zhan ZH, Li J, Cao J, Zhang J, Chung HSH, Shi YH (2013) Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems. IEEE Trans Cybern 43(2):445-463

Zhao J, Liu S, Zhou M, Guo X, Qi L (2018) Modified cuckoo search algorithm to solve economic power dispatch optimization problems. IEEE/CAA J Autom Sinica 5(4):794-806

Zou J, Deng Q, Zheng J, Yang S (2020) A close neighbor mobility method using particle swarm optimizer for solving multimodal optimization problems. Inf Sci 519:332-347

Zou D, Li S, Kong X, Ouyang H, Li Z (2018) Solving the dynamic economic dispatch by a memory-based global differential evolution and a repair technique of constraint handling. Energy 147:59-80