

Democracy-inspired particle swarm optimizer with the concept of peer groups

Ritambhar Burman¹ · Soumyadeep Chakrabarti¹ · Swagatam Das²

Published online: 18 January 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract This article proposes to integrate the concept of governance in human society with the nature-inspired particle swarm optimization (PSO) algorithm. A population-based iterative global optimization algorithm, called Democracy-inspired particle swarm optimization with the concept of peer groups (DPG-PSO) has been developed for solving multidimensional, non-linear, non-convex, and multimodal optimization problems by exploiting the concept of the new peer-influenced topology. Here the particles, each of which model a candidate solution of the problem under consideration, are given a choice to follow two possible leaders who have been selected on the basis of a voting mechanism. The leader and the opposition have their influences proportional to the total number of votes polled in their favor. A detailed empirical study comprising tuning of DPG-PSO parameters and its optimizing mechanism has been presented in the paper. The algorithm is tested in a standard benchmark suite consisting of unimodal, multimodal, shifted and rotated functions. DPG-PSO is found to statistically outperform seven other well-known PSO variants in terms of final accuracy and robustness over majority of the test cases, thus, proving itself as an efficient algorithm.

Keywords Global optimization · Continuous optimization · Particle Swarms · Democracy · Peer group topology

1 Introduction

In 1995, the idea of function-optimization was introduced with the help of a swarm of conceptual particles that fly through the parameter space searching for the optima (Kennedy and Eberhart 1995; Eberhart and Kennedy 1995). Currently, the basic particle swarm optimizer (PSO) and its variants constitute one of the most well-known families of global optimizers over the real or continuous parameter space. In PSO, each candidate solution is modeled as a particle and several such particles together form a swarm. Particles fly through the multidimensional search space following a typical dynamics. At any instance, each particle has a position and a velocity. Initially, a population of particles is initialized with randomly chosen position-vectors and velocities. Each particle then adapts its search pattern by learning from its own experience as well as the experience of the others. The aim of each particle is to move into a better region of the search space with a definite velocity influenced by the information collected by its own self and the other members of the swarm throughout its lifetime. PSO does not require any gradient information to optimize a numerical objective function. It is simple and only uses elementary mathematical operators. Since 1995, PSO has been extensively investigated by the researchers and this has resulted into nearly uncountable number of variants of the algorithm. Dynamics of the particles and parameter selection have been investigated both theoretically and empirically. PSO has been applied to a wide range of real-world problems ranging from diverse domains. For comprehensive surveys on the research on and with PSO, the readers are directed to the references such as (Eberhart

Communicated by V. Loia.

✉ Swagatam Das
swagatam.das@ieee.org; swagatam.das@isical.ac.in

Ritambhar Burman
ritambharkcburman@gmail.com

Soumyadeep Chakrabarti
soumyadeepchakrabarti@gmail.com

¹ Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata 700032, India

² Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata 700108, India

and Shi 2004; Engelbrecht 2006; Clerc 2008; Valle et al. 2008; Liang et al. 2006; Banks et al. 2007; Poli 2008).

The main drawback of PSO, being a stochastic process, is false and/or premature convergence, especially for multimodal functions. Multifariousness is lost as the globally best member influences the dynamics of the particles in the swarm. Premature convergence occurs when a globally best particle, at one of the local optima, traps the population. This had led researchers to propose alternate variants and modifications, which are mostly algorithmic and leads to better performance. The modifications include tuning of selection parameters to achieve a better exploration/exploitation trade-off (Shi and Eberhart 1999; Ratnaweera et al. 2004; Clerc and Kennedy 2002), designing various proximity topologies to replace the traditional global topology (Kennedy 1999; Suganthan 1999; Kennedy and Mendes 2002; Kennedy and Mendes 2006), using multiswarm techniques (Nasir et al. 2012; Van den Bergh and Engelbrecht 2004; Liang and Suganthan 2005; Lovbjerg et al. 2001), and hybridizing auxiliary search methods with PSO (Chen et al. 2007; Andrews 2006; Higashi and Iba 2003; Zhang and Xie 2003; Shelokar et al. 2007; Das et al. 2005). PSO has also been modified by using deterministic sampling such as in the species-based optimization algorithm (Cho et al. 2011) and using simplex search techniques in unconstrained optimization problems (Fan and Zahara 2007). However, majority of such variants preserve the population diversity at the cost of slow convergence or complicated algorithmic structures. Avoiding premature convergence without greatly reducing the convergence speed and without making the algorithmic structure too complicated still remains a challenge for the PSO researchers. The findings reported in this paper can be considered as a humble attempt to eradicate some of these problems.

This article presents a new variant of nature-inspired PSO algorithm that exploits the concept of governance in human society. The new technique is called Democracy-Inspired Particle Swarm Optimizer with the concept of Peer Groups (DPG-PSO). Two leaders are selected on the basis of votes, and these leaders have influence over the particles who vote for them. This technique can competitively solve multidimensional, non-linear, non-convex, and multimodal optimization problems exploiting the concept of the new peer-influenced topology. DPG-PSO is tested against standard benchmark functions and is statistically found to be more robust and accurate than seven other PSO variants.

The rest of the paper is organized in the following way. Section 2 briefly explains the background of PSO along with the concept of democracy and peer group. Section 3 introduces the democracy-inspired optimization technique outlining the basic steps. Section 4 presents and discusses the experimental comparisons of the proposed method with other well-known PSO variants like Ageing Leader Challenger Particle Swarm Optimization (ALC-PSO) (Chen et al.

2013), Fully Informed Particle Swarm (FIPS) (Mendes et al. 2004), Dynamic Multi-Swarm Particle Swarm Optimization (DMS-PSO) (Liang and Suganthan 2005), Orthogonal Learning Particle Swarm Optimization (OLPSO) (Zhan et al. 2010), Comprehensive Learning Particle Swarm Optimization (CLPSO) (Liang et al. 2006), Hierarchical Particle Swarm Optimization with Time Varying Acceleration Coefficients (HPSO-TVAC) (Ratnaweera et al. 2004), and GPSO (Shi and Eberhart 1998) and duly discusses the experimental results. Section 5 discusses on the control parameters of the proposed technique to show its robustness. Section 6 compares the democratic and undemocratic approaches for the DPG-PSO algorithm to show the superiority of the proposed democracy-inspired algorithm. The absence of center-seeking bias of the algorithm is illustrated in Sect. 7 with the help of the experimental results on shifted functions. Finally conclusions are drawn in Sect. 8.

2 Background

2.1 Basic PSO and related works on PSO

PSO, as the name suggests, uses a swarm of particles each of which model a candidate solution of the problem at hand. A particle is characterized by its position vector $x_i = \{x_i^1, x_i^2, \dots, x_i^D\}$ representing a point in the D -dimensional real space (\mathbb{R}^D), and its velocity vector $v_i = \{v_i^1, v_i^2, \dots, v_i^D\}$. The velocity and position of a particle are updated in the following way:

$$v_i^d \leftarrow v_i^d + c_1 * rand1_i^d * (pbest_i^d - x_i^d) + c_2 * rand2_i^d * (gbest_i^d - x_i^d), \quad (1)$$

$$x_i^d \leftarrow x_i^d + v_i^d, \quad (2)$$

where x_i^d and v_i^d , respectively, represent the d -th component of the position and velocity of the i -th particle. The best position of the particle, i.e., the position at which the i -th particle yields its best fitness value is termed $pbest_i$ where $pbest_i = \{pbest_i^1, pbest_i^2, \dots, pbest_i^D\}$. Using similar terminology the global best position of the i -th particle is given by $gbest_i = \{gbest_i^1, gbest_i^2, \dots, gbest_i^D\}$. Note that for the $gbest$ PSO topology $gbest_i$ denotes the best position found so far in the entire swarm and the subscript i is not necessary. On the other hand for the $lbest$ PSO model, $gbest_i$ denotes the best position found by some particle in a neighborhood of the i -th particle. There are different types of neighborhood topologies, namely the ring, pyramid, Von-neumann, and others. Ring neighborhood topology has not only been extensively used in PSO (Kennedy and Mendes 2002), but also implemented in the differential evolution (DE) family of algorithms, see for example (Omran et al. 2006; Das et al.

2009). Without loss of generality, our method can be integrated with both the *gbest* and *lbest* PSO models with any proximity topology.

The constant c_1 scales the attraction of a particle towards the best position (hitherto-optimum position) it has found so far. On the other hand, the constant c_2 does the same with the best position found so far by the entire swarm or in a neighborhood of the current particle. These are often called acceleration coefficients. Venter and Sobieski (2003) termed ‘ c_1 ’ as self-confidence and ‘ c_2 ’ as *swarmconfidence*. $rand1_i^d$ and $rand2_i^d$ are uniformly distributed random numbers bounded within the range [0,1] and are instantiated freshly for each ordered pair (d, i) . The velocity of a particle is limited to a maximum denoted by $v_{\max} = [v_{\max}^1, v_{\max}^2, \dots, v_{\max}^D]$. If the velocity exceeds this limiting condition, appropriate steps are taken to reduce it. Reassignment of the velocity component to a value of $sign(|v_i^d|)v_{\max}^d$ remains a popular means of limitation.

Parameter control and adaptation have attracted the PSO researchers for a long time. The inertia weight ω was first introduced by Shi and Eberhart (1998) to dampen the inertial velocity from previous iteration and thus to influence convergence. They showed that while larger inertia weight is advantageous in wide-scale exploration, a smaller one may increase the ability of local refinement. Since then, almost all the PSO variants have used the inertia weight as an integral part of the velocity update rule. To balance the local and global search abilities, Shi and Eberhart put forth a scheme to decrease ω linearly with number of iterations in the following way:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \cdot \frac{g}{G}, \quad (3)$$

where g is the iteration index and G is a predefined maximum number of iterations. ω_{\max} and ω_{\min} are usually set as 0.9 and 0.4, respectively. Thus, the use of time varying inertia weight ω provides the necessary balance between the local and global search abilities of a particle. A fuzzy rule-based non-linear adaptation of ω was proposed by Shi and Eberhart (2001). In context to the dynamic system optimization, a significant modification of inertia weight as $\omega = 0.5 + rand(0, 1) * 5$ was experimented (Eberhart and Shi 2001). Logarithmic and exponential decreases of the inertia weight have been investigated in Eberhart and Shi (2001); Liao et al. (2012). Quande et al. presented a novel PSO with a piecewise-varied inertia weight. The piecewise function thus chosen is a decreasing function as both its parts are so. The first part of the piecewise function thus chosen is non-linear which helps in the search for the global optimum and restricts premature convergence. The second part is linear and found in the conventional PSO (Shayeghi et al. 2010). A variation for realizing the convergence behavior of the particles is by the use of a constriction factor χ introduced by Clerc and Kennedy (2002) in the following way:

$$v_i^d \leftarrow (v_i^d + c_1 * rand1_i^d * (pbest_i^d - x_i^d) + c_2 * rand2_i^d * (gbest_i^d - x_i^d)), \quad (4)$$

where $\chi = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}$ and $\phi = c_1 + c_2$.

Besides the inertia weight and constriction factor, acceleration coefficients can greatly influence the performance of PSO as indicated by the “social only” and “cognitive-only” models (Quande et al. 2010). A fixed value of 2 as suggested in Kennedy and Eberhart (1995), Eberhart and Kennedy (1995) is mostly preferred by the researchers for both the coefficients. Time Varying Acceleration Coefficients (TVAC) employed in a PSO variant called HPSO (Hierarchical PSO)-TVAC (Ratnaweera et al. 2004) and the use of ad hoc values of acceleration coefficients are some of the modifications that highlight the influence of acceleration coefficients on the performance of PSO. In *lbest* PSO models, due to the neighborhood structures, the particles show better resistance towards trapping in local optima. But the disadvantage of these models is the deficient leadership capability in the swarm as a result of which many particles exert greater influence on any one particle, resulting in a de-centralized behavior. Such PSO variants usually exhibit a slow convergence characteristic. In context to the *lbest* PSOs, a multitude of neighborhood topologies has been investigated; see for example works like (Kennedy and Mendes 2002). Usually a large neighborhood proves to be effective for simpler optimization problems, while a restricted neighborhood is better for complicated problems with multiple modes. Topologies which change over time have also been researched and these have been shown to be robust on various functional landscapes (Suganthan 1999; Kennedy 1997). Fully Informed Particle Swarm (FIPS) algorithm was proposed by Mendes et al. (2004). FIPS adjusts the particle velocity by taking into account the positions found in other neighborhoods in addition to considering the particle’s neighborhood optimum. The same idea was used in conjunction with Euclidean neighborhoods to induce significant niching behavior in PSO (for the detection of multiple optima simultaneously on a multimodal fitness landscape) by Qu et al. (2012).

Even the particle in the optimum position may not have the best fitness when all the dimensions are concerned. The swarm should benefit from utilizing the best possible position along every dimension separately and this was the inspiration for the Comprehensive Learning PSO (CLPSO) which gave better results because of the increase in swarm diversity. Randomized variable neighborhoods are used in the standard PSO model (SPSO’07) (Bratto and Kennedy 2007). Janson and Middendorf proposed a hierarchical version of PSO (HPSO) (Janson and Middendorf 2005) that arranges the particles in a hierarchy, which defines the neighborhood structure. Every particle has itself and its parents in the system as neighbors.

Several efforts have been made to devise an improved hybrid algorithm by synergizing PSO with other local or global optimization techniques. Incorporating the operators of a Genetic Algorithm (GA) inside the conventional PSO framework, such as selection (Angeline 1998), crossover (Chen et al. 2007), and mutation (Andrews 2006; Higashi and Iba 2003), have led to efficient optimization in many instances. Several attempts have been made to hybridize the features of PSO with another very powerful real-parameter optimizer called Differential Evolution (DE) and a detailed account can be found in Xin et al. (2012). Apart from synergy with other algorithms, deflection, stretching, and repulsion techniques (Parsopoulos and Vrahatis 2004) were combined with PSO to improve its performance. Mutation strategies for parameters like inertia weight (Miranda and Fonseca 2002), diversity maintaining scheme, as well as distance-dependent schemes to prevent particles from congestion in a particular area of the search space (Lovbjerg and Krink 2002; Blackwell and Bentley 2002) have also been proposed for PSO.

Some researchers attempted to integrate the concepts of aging and lifespan in PSO framework for obtaining improved search on multimodal landscapes. A trapezoidal aging function (Dehuri et al. 2006) was used to mark the lifespan of particles in context to data clustering. A PSO with aging leaders and challenger particles has been recently proposed (Chen et al. 2012). Multipopulation approaches (also more familiarly called multiswarms) (Tillett et al. 2005) have received considerable attention from the PSO researchers over the past few years. In case of PSO variants with multiple sub-populations, the sub-populations themselves can be treated as a special type of neighborhood structure. Considering this fact, such category of PSO variants can also suffer from slower convergence speed. Recently, a very competitive variant of PSO was proposed (Zhan et al. 2011) by using an orthogonal learning strategy to discover more useful information that lies in the learning experiences of a particle via the orthogonal experimental design (OED).

Different types of topologies have been used to improve the performance of PSO. Some PSO algorithms use a fully connected neighborhood topology. The best solution found by any particles acts as the focal point of aggregation for all particles. Each particle has information concerning all the particles in the swarm. On the other hand, there are also topographies where every particle has knowledge pertaining to only a restricted neighborhood. One common topology is the ring topology (Kennedy and Mendes 2002), in which each and every particle is connected with just two neighbors. Kennedy claimed that PSO with a small neighborhood might perform better on complex problems, because it slowly searches for the optimum while PSO with a large neighborhood would perform better on simple problems where it will converge faster albeit with a higher chance of premature convergence, if possible for that function. A

dynamically adjusted neighborhood topology (Suganthan 1999) was introduced with a gradually increasing neighborhood. Hu and Eberhart (2002) adopted a topology where nearest particles on the basis of performance are included in its new neighborhood in each generation. In fully informed particle swarm (FIPS) proposed by Mendes and Kennedy, each individual is influenced by the differently weighted successes of all its neighbors, rather than just the best one and itself. A particular problem may be better handled by one structure but some other application might require another structure to be correctly solved.

New adaptive techniques are introduced to classical particle swarm optimization to give an adaptive two-layer particle swarm optimization algorithm with elitist learning strategy (ATLPSO-ELS) (Lim and Isa 2014), which has better search capability than classical. In ATLPSO-ELS, both the current swarm and the memory swarm are evolved in a certain fashion that the tendency of the latter swarm to distribute around the problem's optima. Better control of exploration/exploitation searches in both current and memory swarms is achieved by two adaptive divisions of labor modules to self-adaptively divide the swarms into explorative and exploitative groups. Additionally, OED and stochastic perturbation techniques are used to develop modules to enhance the search efficiency of swarms and to mitigate premature convergence.

Traditional PSO consisting of only two searching layers is also replaced by new multilayer particle swarm optimization (MLPSO) (Wang et al. 2014) to overcome premature convergence into the local minima. The MLPSO strategy increases the diversity of searching swarms to efficiently solve complex problems.

DPSO (Kaveh and Zolghadr 2013) was the first optimisation technique which was inspired by principles of democracy. But its scope as well as implementation is markedly different from the technique proposed in this paper. Principally, as far as the algorithm is concerned, in DPSO, democracy is implemented by engaging every particle in decision making directly, whereas in DPG, the particles enjoy the right to select their own respective leader and exercise their democracy in an indirect manner through this leader. In DPSO, there is an extra term in the velocity update equation which is a direction vector constructed on the basis of the entire population while in DPG, there is no extra term in the velocity update equation. Additionally, in DPSO, the basic topology remains the same as in basic PSO, but in DPG a new variant of the ring topology, referred to as the peer-based topology, is used.

2.2 Concepts of democracy and peer group algorithm

Basic PSO introduced the concept of leader (the globally best particle) and it has already proved its importance. How-

ever, single leadership does not always ensure a better result in optimization algorithms and can lead to unwanted premature convergence. This realization calls for the introduction of democratic form of government. In the present day, democracy is based on the people electing people from their midst who take the decisions and perform actions which can be potentially beneficial to the state as a whole, thus making democracy a form of governance by the people, of the people, and for the people. Democracy also allows the concept of cross fertilizations of ideas between people which are beneficial to them, thus true democracy is seen when people gain knowledge and advance together (Fishkin 2005).

Added advantage of democracy is that it checks the manipulation of the total masses by one leader and introduces in real sense the concept of *Opposition*. Choice is essential in any democratic structure. The population should always have the benefit of a competent alternative to the ruling party which can replace the incumbent leader and lead the population into a more advanced developmental state. The presence of the *Opposition* forces the leader to perform and in the absence of any competition the leader may not have enough inspiration to perform as much as he/she can or should.

Without inherent stability and unity in a society, no leader can guide it to its goals. Thus, each individual is influenced by his/her friends and families to a large extent even though individuals in their friend and family circle may not have exceptionally strong leadership qualities. This realization leads to the introduction of a new topology based on the concept of peer group or peer circle in the PSO system.

In this paper, the focus is on exploiting the human society concept of the freedom in governance and peer influence. In basic principle, this novel strategy allows each particle to choose their leader by using the concept of democracy. This shifts from the previous use of democracy concept in PSO (Kaveh and Zolghadr 2013; Kaveh 2014) in that the choice of the people is carried out indirectly through the leader, rather than directly as reflected by the direction vector in the velocity update equation in DPSO. The particles use a closed topology where they interact with only certain nearby particles which form their respective peer groups. As the result, the particles properly update themselves by choosing the proper leader for themselves and by taking proper inputs from its peer circle.

3 Democracy-inspired particle swarm optimization with concept of peer group (DPG-PSO)

3.1 Procedure for DPG-PSO

The original PSO uses one leader of the swarm; however, DPG-PSO uses two leaders. One of the leaders is as its basic *Governor* who leads a majority of the particles and the other

acts as the *Opposition* who leads certain group of particles and always checks the power and sees to whether the leader is leading the society towards the optimum position. The selection of the leader is done by a voting mechanism where each particle votes for the *Governor* or *Opposition* and selects them as their leader to lead them for the particular run. The concept of peer group states that the particles should consider its *lbest*. Thus the concept of *gbest* in globally best topology PSO is modified with the introduction of the concept of leader with both *Governor* and *Opposition* having a chance of leading the society. Hence, the misleading effects which occur due to *gbest* are expected to be reduced.

Thus the velocity and position update equation is modified to

$$v_{ij} \leftarrow \omega * v_i^d + c_1 * rand_1 * (pbest_i^d - x_i^d) * (1 - \frac{FE}{FE_{max}}) + c_2 * rand_2 * (lbest_i^d - x_i^d) * (1 - \frac{FE}{FE_{max}}) + c_3 * rand_3 * (Leader_i^d - x_i^d) * \exp(1 - vote_{leader}), \tag{5}$$

$$x_i^d \leftarrow x_i^d + c_4 * rand_4 * v_i^d. \tag{6}$$

3.2 Modified velocity update equation

The original PSO velocity equation has been modified to include the ideas of peer group and democracy as shown in (5) and (6). The velocity update equation now contains four terms:

- 1) Inertia factor
The concept of inertia or the particle’s tendency to follow its velocity is here as same as that of basic PSO equation, given by $\omega \times v_i^d$.
- 2) Personal best
The distance of a particle’s current position x_i^d from its personal best $pbest_i^d$ is calculated. This is multiplied by acceleration factor c_1 and a uniform random number $rand_1 \in [0, 1]$. However to ensure faster convergence, this term is multiplied with another factor that controls the impact of this term with time and slowly reduces it. After each iteration, a particle is expected to reach properly to its optimum position. Consequently the distance between the $pbest_i^d$ and x_i^d is expected to decrease and thus, this term fails to update the velocity much. Therefore the impact of this term is made to diminish with time using the expression:

$$c_2 * rand_1 * (pbest_i^d - x_i^d) * (1 - \frac{FE}{FE_{max}}).$$

- 3) Local best

Each particle searches its peer circle and finds the best position among them. Encompassing the concept of peer group, this term tends towards its own local best position. Thus the difference between particle's current position x_i^d from its local best $lbest_i^d$ is calculated and multiplied by another acceleration factor c_2 and a random real number $rand_2$ uniformly distributed within $[0, 1]$. Here also to ensure fast convergence we slowly reduce the impact of this term using the expression:

$$c_2 \times rand_2 \times (lbest_i^d - x_i^d) \times \left(1 - \frac{FE}{FE_{max}}\right).$$

4) Leader factor

This term embodies the concept of democracy and also gives significance to the votes of the people. Leader is chosen as either the *Governor* or the *Opposition* through votes. The particles are attracted to their voted *Leader* and tend to move towards it. Thus, the difference between $Leader_i^d$ and x_i^d is taken and multiplied by new coefficient c_3 called attraction coefficient and a uniform random number $rand_3 \in [0, 1]$. It is seen that with passage of time $oVote$ decreases and $gVote$ goes on increasing due to expected better leadership quality of *Governor*. Thus, the influence of *Opposition* starts weakening. This although good for much faster convergence but can hamper the convergence if the *Governor* who starts leading a majority of mass gets caught up in local minima. It is at time as these that the *Opposition* is expected to lead people to the global optima. Thus the effect of those following *Opposition* should be strengthened even if $oBest$ is very less. Thus an empirically developed expression is multiplied with this term to fight such scenario.

$$c_3 \times rand_3 \times (Leader_i^d - x_i^d) \times \exp(1 - vote_{Leader}).$$

Voting is the most essential factor of democracy. To implement voting mechanism a mathematical model of a biased roulette wheel scheme is used where the asymmetric ranges corresponding to the two candidates are constructed by dividing the range $[0, 1]$ into two parts each of which is proportional to the number of votes of the respective candidate namely $gVote$ and $oVote$ calculated by the normalization procedure. The need for initially biasing the range is that initially the population is generated with random values hence they lack the ability to decide on their own which *Leader* can lead them better. Hence initially the best particle from the population is selected as *Governor* and second best particle as *Opposition* and $gVote$ is given a value Φ and the $oVote$ is set to $1 - \Phi$. This initial value Φ bias the votes. However in consecutive runs each particle for each dimension poll a vote which is a random number generated within the range $[0, 1]$. Based on this vote its *Leader* is selected in the following manner:

$$Leader_i^d = \begin{cases} Governor^d, & vote_i^d \leq gVote \\ Opposition^d, & vote_i^d > gVote \end{cases} \quad (7)$$

Once the *Leader* is selected the particles update the value of $gVote$ and $oVote$ to show their *Leader* is more preferred than the other. The vote count of each *Leader* is increased with one part of $N * M$. Thus along with the *Leader* selection the vote is updated in the following manner.

$$\begin{aligned} \text{If } vote_i^d < gVote, & \quad gVote \leftarrow gVote + \frac{1}{N * M} \\ \text{Else, } & \quad oVote \leftarrow oVote + \frac{1}{N * M}. \end{aligned} \quad (8)$$

The votes polled in favor of any candidate are taken in a cumulative format. This helps in realizing which *Leader* has throughout the time acted as a better *Leader*. Thus votes got at the end of each iterations are normalized using (9), (10) and carried forward for the next iteration.

$$gVote \leftarrow \frac{gVote}{gVote + oVote} \quad (9)$$

$$oVote \leftarrow \frac{oVote}{gVote + oVote} \quad (10)$$

$vote_{Leader}$ is the normalized number of votes received by a particular leader, be it the *Governor* or the *Opposition*. The importance of the empirical expression $\exp(1 - vote_{Leader})$ is seen in the following way:

Case 1: $Leader_i^d \leftarrow Governor_d$. In this case, $vote_{Leader} \leftarrow gVote$ which almost tends to 1 with passage of time results in $\exp(1 - gVote) \approx \exp(0) \approx 1 \forall gVote \rightarrow 1$. Thus it does not alter the impact for *Governor* much if he has taken majority into confidence or the manner he used to influence them.

Case 2: $Leader_i^d \leftarrow Opposition_d$. In this case, $vote_{Leader} \leftarrow oVote$ and if $oVote$ is very small then $\exp(1 - oVote) \approx \exp(1) \approx 2.3 \forall oVote \rightarrow 0$. Thus the *Opposition* will still be able to influence the people who vote for him and in case when the *Governor* fails and leads the majority of swarm to local optima. In such situations, *Opposition* leads his believers and slowly the whole swarm to the global optima. The voting mechanism is illustrated under biased roulette scheme in Fig. 1.

3.3 Modified position update equation

The basic position update equation of original PSO has also been changed to introduce certain randomness and weight factor to the velocity term. c_4 is the velocity scaling factor. $rand_4$ is a uniform random number within the range $[0, 1]$.

In DPG-PSO, a particle does not change its position depending largely velocity term. Here its current position is made more dominant than its velocity term by scaling down the latter using c_4 which is taken to a value less than 1. To

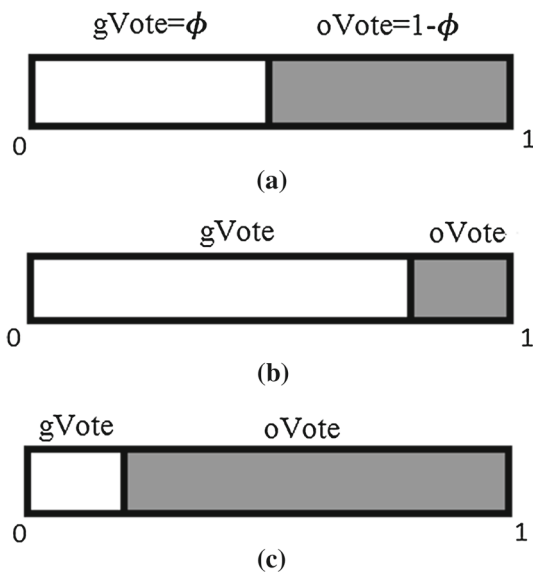


Fig. 1 Voting mechanism under biased roulette scheme. **a** shows the division of range [0,1] in the initial run. **b** Shows the distribution of range when the *Governor* is better Leader and hence *gVote* increases at the cost of *oVote*. **c** Shows the distribution of range when the *Opposition* is better Leader

understand the dynamics behind the above point let us take a scenario where a particle has already reached global best, however other particles are still distributed far apart from him. Due to the tendency of the particle to move towards the swarm it might get dislocated from the global optima. Due to c_4 the position term will try to dominate over the velocity term and keep particle at the reached global optima. The impact of velocity term has been stochastically varied for different dimension to enhance the explorative nature of particles using $rand_4$.

3.4 Peer-based topology

The learning process of particles in any PSO is highly dependent on the topology and how they interact in the defined neighborhood structure. The basic PSO has a global neighborhood (or mesh topology) where every particle can learn from the best particle in the entire swarm. This results in faster convergence but increases the liability of the swarm to be trapped in a local optimum. The explorative nature of the swarm can be enhanced by using a topology where the neighborhood of a particle includes a few individuals rather than only one particle such as the Ring topology.

An example of a simple ring topology defined on the index graph of the particles is illustrated in Fig. 2. We assume there are N particles to be organized on a ring topology with respect to their indices, such that particles $P_N(t)$ and $P_2(t)$ are the two immediate neighbors of particle $P_1(t)$. Only the immediate neighbors play a role in the position and velocity update equations of any particle.

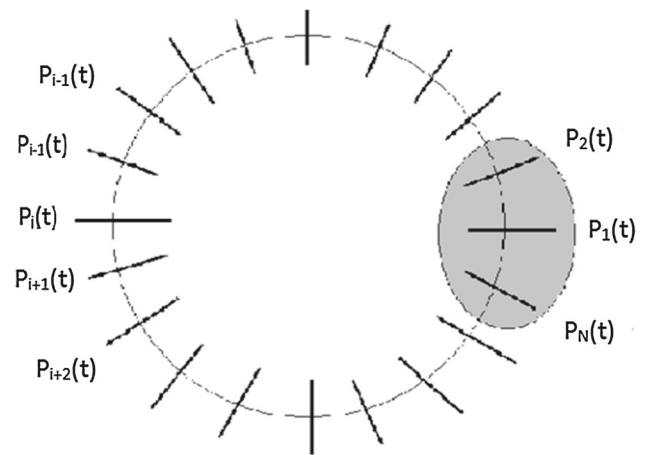


Fig. 2 Diagram showing the ring neighborhood topology in PSO. The shaded region indicates a neighborhood of the i^{th} particle containing two other particles when $i = 1$ and swarm size N

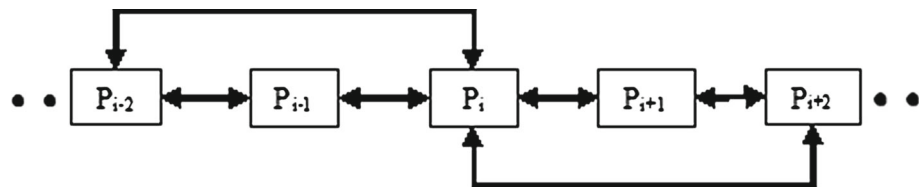
Drawing inspiration from ring topology we introduce a new topology called peer-based topology in this paper where particles interact not only with the immediate two neighbors as in ring topology but certain particles beyond. Here the neighborhood size is 2 and the particles are arranged in a topology as shown in Fig. 3.

A partly connected topology is one where the population does not have a global best. An influential variant of ring topology is one where diffusion of possible knowledge of optima is better conducted than in the basic ring topology of PSO. A trade-off between the partly connected and influential topology is necessary for a democracy-inspired voting-based PSO so that the populism scheme can be successfully implemented in the swarm of the particles without any particular *Leader* being disproportionately influential.

In a ring topology all the particles may be thought to form a ring where the neighborhood size is 2. In the proposed topology the basic ring structure is altered by changing the neighborhood size from the constant value of 2. The neighborhood varies from 2 to 3 depending on the particle dynamics.

To understand how this new topology works let us see how $lbest$ is updated by using this topology in contrast with the ring topology. In ring topology structure, the local best of each particle is updated by computing the cost function of the particle i and comparing it with the cost value of $lbest$ of its neighboring particles $i + 1$ and $i - 1$. If cost value of particle i is lower than $lbest$ of its neighboring particles, then particle i assumes the $lbest$ position of the respective particle $i + 1$ or $i - 1$. In peer-based topology, in addition to the ring topology-based interaction, each of the particles i can also assume the $lbest$ position of its immediate neighbors, if its cost function is lower than the cost function of $lbest$ of the particles $i + 2$ or $i - 2$. This results in a fundamental difference from basic ring topology in a particular case: Suppose that the

Fig. 3 Diagram showing interaction between Particle $i (P_i)$ and its neighbors in peer-based topology



$lbest$ of particle $i + 1$ or $i - 1$ is the $pbest$ of particle i , and this $pbest$ has a lower cost value than present cost value of particle i . Even then the particle i is capable of assuming the $lbest$ position of particles $i + 1$ or $i - 1$, if it manages to have a lower cost function than that of $lbest$ of particles $i + 2$ or $i - 2$. Thus a particle can influence the behavior of its neighbors even though it might be weaker than its own personal best, reflecting the nature of peer-influenced human societies.

To illustrate the concept of the peer-based topology let us consider a minimisation problem. Five connected particles (two on either side of the central particle i) among all the population is shown in Fig. 3 having hypothetical cost values of their $lbest$ and $pbest$ positions. Also shown is the (hypothetical) cost value of current location of particle i , on which the algorithm is stuck on at any particular time. Simplified values are given for representative purposes and the minimum is assumed to be at 0, such that smaller cost value corresponds to better fitness.

Thus particles $i - 2$ and $i + 2$ change their $lbest$ values under the influence of particle i 's new current value. This is because cost value of $lbest$ of particle $i - 2$ is 2 and cost value of $lbest$ of particle $i + 2$ is 2.5, both of which are less than current cost value of particle i , that is 1.5. The updated values of $lbest$ are shown in Fig. 4. And this updating action is indicated by the arrows in Fig. 4.

3.5 Generation of opposition and its update equation

Opposition is generated by imbibing certain properties from best particle of the swarm (*Governor*) and also having some of its random properties. The basic aim while generating the *Opposition* is to introduce certain randomness in its properties so that he is not a replica of *Governor* (the best particle).

This guarantees that the *Opposition* is different from *Governor* and he can lead the particles to better result when the *Governor* fails to do so.

While generating *Opposition*, a random number rnd_d uniformly distributed within $[0, 1]$ is generated for each dimension and is compared with a parameter pro . If $rnd_d < pro$, then the $Opposition_d$ is set to random real number within the range $[X_{min}, X_{max}]$ else $Opposition_d$ is taken from best particle.

$$Opposition_d = \begin{cases} rand \times (X_{max} - X_{min}) + X_{min}, & rnd_d < pro \\ Governor_d, & rnd_d \geq pro \end{cases} \tag{11}$$

where $d \in \{1, 2, 3, \dots, N\}$

To prevent *Opposition* having too much random properties so that the convergences takes a lot of time, the parameter pro is taken to be small and equal to $1/N$ empirically.

Opposition updates itself in each run in the same stochastic manner as it is generated; by learning from the best particle of swarm at the end of each iterations.

The flowchart of DPG-PSO is illustrated in Fig. 5 and the steps involved are given as follows.

Step1 Initialization: The initial positions and velocity of all the particles are randomly generated within the n -dimensional search space. The best particle is selected as *Governor* and *Opposition* is formed by process of mutations. The vote for the *Governor* $gVote$ and the *Opposition* $oVote$ is set to initial value to control voting mechanism using biased roulette system.

Step 2 Selecting the *Leader* using voting system: Each particles generate a random number within the range $[0, 1]$ which acts as its vote. Based on this vote, its *Leader* is selected.

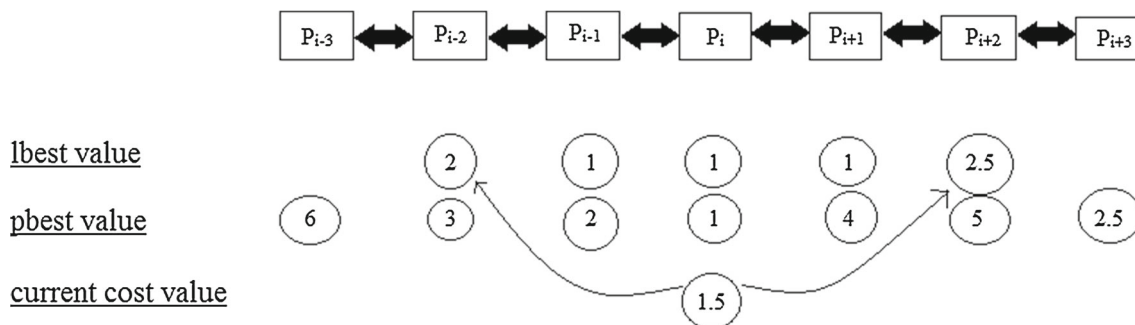


Fig. 4 Diagram showing a part of the particle population for illustration purposes: After updating cost values

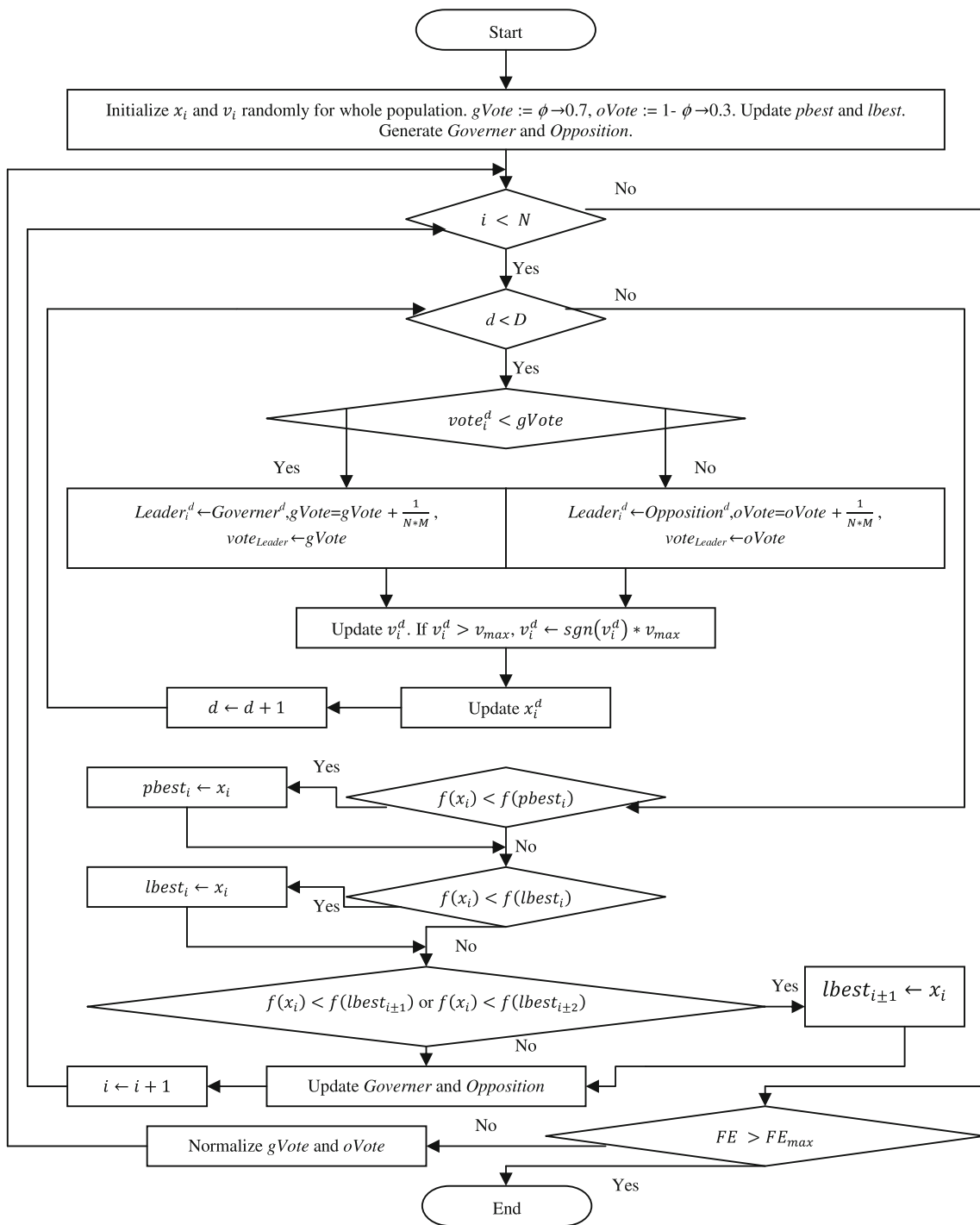


Fig. 5 Flowchart representation of the DPG-PSO algorithm.

Step 3 Velocity and position Updating: Each particle updates its velocity and its position using (5) and (6), respectively.

Step 4 Updating *pbest*, *lbest*, *Governor*, and *Opposition*: If the new position of particle $i (i \in \{1, 2, \dots, M\})$ is better than $pbest_i$, then $pbest_i$ is updated to x_i . After the completion of each iteration, $lBest_i$ is updated if the position of particles in its peer group becomes better than $lbest_i$. *Governor* also

updates itself to become the best position in this iteration and *Opposition* updates itself taking certain properties of the best particle.

Step 5 Updating the votes: The individual votes got in this iteration is added to the previous votes of *Governor* and *Opposition*. In end of each iterations $gVote$ and $oVote$ get normalized with respect to total votes in the iteration.

Step 6 Terminal Condition Check: If the number of FES exceeds x_{FE} , the algorithm terminates. Otherwise go to Step 2 for next iteration.

According to the above procedure, DPG-PSO introduces two new concepts of voting allowing each particle to choose their *Leader* between *Governor* and *Opposition*. It also withdraws the concept of *gbest* and uses *lbest* where the particles interact with two or three of its neighboring particles and finds *lbest*.

4 Experimental verifications and comparisons

4.1 Benchmark function tested

Experimental tests are performed on the eighteen standard benchmark functions listed in Table 2. These functions have been frequently used for the testing of new PSO variants. Table 2 also shows the global optimum value and the search ranges of the defined functions. Since they are minimizing functions, the optimum is the global minima for the problems. These benchmark functions are tested on seven well-known variants of PSO like GPSO, FIPS, HPSO-TVAC, DMS-PSO, ALCPSO, CLPSO, and OLPSO. The eighteen benchmark functions are classified into three categories, the first group $f_1 - f_6$ having six unimodal functions. The second group has four multimodal functions $f_7 - f_{10}$ which are more complex with higher dimensionality and multiple local optima. The last group $f_{11} - f_{18}$ contains 3 rotated and 5 shifted functions. For the details of the shift vector and the orthogonal rotation matrices corresponding to functions $f_{11} - f_{18}$, see (Suganthan et al. 2005).

4.2 Experimental setup

The experimental results tabulated in Table 3 are obtained by running these algorithms on the benchmark functions of Table 2, are done with parameter settings indicated in Table 1. All the algorithms are tested for 30 dimensions with a swarm size of 20 particles for the PSO variants. To remove

statistical dependence, the algorithms are independently run 30 times each and their results are tabulated in Table 2. The simulation environment is Matlab 2013a in a workstation with Intel Core i5 and 2.50 GHz Processor. For the competitor algorithms, we use the same parametric settings as mentioned in the respective literature. The parametric setup for the compared algorithms has been provided in Table 1.

4.3 Experimental results

DPG-PSO algorithm and the other well-known variants of PSO are tested on the eighteen benchmark functions $f_1 - f_{18}$ of Table 2. Table 3 gives the mean error, standard deviation and least error value over a period of 30 runs to remove statistical dependence. Ranking has been done according to the mean error value obtained by the algorithms. For minimization problems, the best error value indicates the run securing the least error among the 30 runs for a particular algorithm. In Table 3, the best results comprising mean error and the least error are marked in bold face for each of the functions described in Table 2. Error value is defined as $|f(\mathbf{x}_{\text{best}}) - f(\mathbf{x}_{\text{opt}})|$ where \mathbf{x}_{best} is the best solution returned by DPG-PSO and \mathbf{x}_{opt} is the global minimum position of the corresponding function. The functions are tested with 30 dimensions, a swarm size of 20 and exhaustion of 2,00,000 function evaluations (FEs) is taken as terminating criteria. If results of a particular test algorithm are unavailable for certain benchmark functions, N/A or not available is written. Average rank is next calculated by the mean of all the ranks that could be evaluated for the benchmark functions for all the algorithms to test the overall performance of the algorithms.

4.4 Comparison

1. Unimodal functions:

Table 3 shows for the six unimodal benchmark functions $f_1 - f_6$ of Table 2, DPG-PSO secures the 1st rank for 5 unimodal functions, and secures the 8th rank for Rosenbrock's function f_5 . Convergence plots of Fig. 6a, c, e, g,

Table 1 Parameter settings of the different variants of PSO used (meaning of the symbols are provided in the respective literatures)

Algorithm	Parameter
GPSO	$\omega = 0.9 \sim 0.4$; $C_1, C_2 = 2$; $V_{\max} = 0.2 * \text{Range}$
FIPS	$\kappa = 0.9 \sim 0.4$; $\sum c_i = 4.1$; $c_2 = 1.4$; $V_{\max} = 0.5 * \text{Range}$
HPSO-TVAC	$\omega = 0.9 \sim 0.4$; $C_1 = 2.5 \sim 0.5$; $C_2 = 0.5 \sim 2.5$; $V_{\max} = 0.5 * \text{Range}$
DMS-PSO	$\omega = 0.9 \sim 0.2$; $C_1, C_2 = 2$; $m = 3$; $R = 5$; $V_{\max} = 0.2 * \text{Range}$
ALCPSO	$\omega = 0.4$; $C_1, C_2 = 2$; $\Theta = 60, (2, 1, 0, 1)$; $V_{\max} = 0.5 * \text{Range}$
CLPSO	$\omega = 0.9 \sim 0.4$; $C_1 = 1.49445$; $m = 7$; $V_{\max} = 0.2 * \text{Range}$
OLPSO	$\omega = 0.9 \sim 0.4$; $C_1, C_2 = 2$; $V_{\max} = 0.5 * \text{Range}$
DPG-PSO	$\omega = 0.2$; $C_1 = 2, C_2 = 1.5, C_3 = 0.5, C_4 = 0.8$; $\Phi = 0.7$; $V_{\max} = 0.5 * \text{Range}$

Table 2 Details of the benchmark functions

Test function	Optima	Range	Name
$f_1(x) = \sum_{i=1}^D x_i^2$	0	[-100,100]	Sphere
$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	0	[-10,10]	Schwefel's P2.22
$f_3(x) = \sum_{i=1}^D i x_i^4 + \text{random}[0, 1]$	0	[-1.28,1.28]	Noise
$f_4(x) = \sum_{j=1}^j (\sum_{i=1}^i x_i)^2$	0	[-100,100]	Quadric
$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	0	[-10,10]	Rosenbrock
$f_6(x) = \sum_{i=1}^D (\text{floor}(x_i + 0.5))^2$	0	[-100,100]	Step
$f_7(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	0	[-5.12,5.12]	Rastrigin
$f_8(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	0	[-32,32]	Ackley
$f_9(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$	0	[-600,600]	Griewank
$f_{10}(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10], x_i = \frac{\text{rand}(0,2\pi)}{2} i, x_i > 0.5$	0	[-5.12,5.12]	Non-continuous Rastrigin
$f_{11}(x) = \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10], y = M \times x, M$ is an orthogonal matrix Suganthan et al. (2005)	0	[5.12,5.12]	Rotated Rastrigin
$f_{12}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D y_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi y_i)\right) + 20 + e, y = M \times x, M$ is an orthogonal matrix	0	[-32,32]	Rotated Ackley
$f_{13}(x) = \frac{1}{4000} \sum_{i=1}^D y_i^2 - \prod_{i=1}^n \cos(y_i/\sqrt{i}) + 1, y = M \times x, M$ is an orthogonal matrix Suganthan et al. (2005)	0	[-600,600]	Rotated Griewank
$f_{14}(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2 - 450, z = (x - o) \times M$	-450	[-100,100]	Shifted Rotated High Conditioned Elliptic
$f_{15}(x) = \sum_{i=1}^{D-1} [100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2] + 390, z = x - o + 1$	390	[-100,100]	Shifted Rosenbrock's
$f_{16}(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (z_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k * 0.5)]$ +90, a = 0.5, b = 0.3, k _{max} = 20, z = (x - o) × M	90	[-50,50]	Shifted Rotated Weierstrass
$f_{17}(x) = \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10] - 330, z = (x - o) \times M$	-330	[-100,100]	Shifted Rotated Rastrigin's
$f_{18}(x) = \sum_{i=1}^D z_i^2 - 450, z = x - o + 1$	-450	[-100,100]	Shifted Sphere

Table 3 Experimental results tested on benchmark functions showing mean error, minimum error, and standard deviation over a period of 30 runs and 200000 FEs, Wilcoxon's rank-sum test (denoted by *P* values) and average ranking of the algorithms based on mean error

Function	Performance Metrics	DPG-PSO	ALCPSO	GPSO	FIPS	HPSO-TVAC	DMS-PSO	CLPSO	OLPSO
<i>f</i> ₁	Mean	0.00	1.89e-162	7.16e-53	3.1e-30	2.13e-41	2.14e-54	1.52e-27	1.69e-12
	Best	0.00	1.94e-172	1.88e-56	5.2e-31	2.62e-48	2.48e-61	2.27e-29	5.17e-14
	SD	0.00	7.7e-161	1.71e-52	3.7e-30	4.92e-41	8.95e-54	2.859e-27	8.4e-13
	<i>P</i> values	NA	3.13e-12	4.52e-18	4.55e-21	6.38e-19	5.71e-16	8.46e-23	2.73e-15
	Rank	1	2	4	6	5	3	7	8
<i>f</i> ₂	Mean	0.00	1.237e-91	3.12e-34	1.4e-17	1.15e-22	2.14e-27	3.891e-17	5.76e-08
	Best	0.00	1.588e-98	8.79e-38	5.6e-18	2.27e-24	4.31e-34	1.125e-17	1.98e-08
	SD	0.00	4.445e-90	6.23e-34	8.5e-18	2.34e-22	0.87e-26	1.768e-17	5.46e-09
	<i>P</i> values	NA	1.67e-20	6.14e-19	5.22e-14	3.56e-17	2.52e-19	1.67e-24	3.62e-22
	Rank	1	2	3	6	5	4	7	8
<i>f</i> ₃	Mean	6.59e-05	5.51e-04	9.32e-03	4.96e-03	9.48e-02	1.69e-02	5.85e-03	6.95e-03
	Best	3.40e-05	6.89e-04	2.39e-03	1.28e-03	3.26e-02	5.22e-03	1.11e-03	1.58e-03
	SD	3.01e-06	9.81e-05	3.78e-04	8.46e-04	7.41e-04	4.87e-04	7.45e-04	1.02e-03
	<i>P</i> values	NA	8.27e-03	1.72e-04	2.97e-04	3.67e-03	3.52e-04	1.62e-06	3.85e-02
	Rank	1	2	6	3	8	7	4	5
<i>f</i> ₄	Mean	0.00	1.648e-11	5.841e-02	3.28e-01	1.169e-07	23.964	106.71	109.54
	Best	0.00	8.312e-14	3.135e-03	.245e-02	5.00e-09	3.01e-1	31.410	45.14
	SD	0.00	3.501e-11	6.162e-02	1.87e-01	1.471e-07	28.634	52.689	59.46
	<i>P</i> values	NA	4.72e-18	7.82e-27	6.59e-25	3.72e-20	1.19e-25	3.08e-25	7.62e-27
	Rank	1	2	4	5	3	6	7	8
<i>f</i> ₅	Mean	28.055	7.655	25.746	23.059	11.464	19.536	17.231	11.22
	Best	27.871	3.714e-7	1.345e-2	21.378	9.876e-6	3.687e-1	9.145e-1	8.76e-7
	SD	0.2931	6.163	26.023	0.439	16.450	12.019	13.451	9.19
	<i>P</i> values	3.51e-06	NA	7.51e-04	2.31e-06	6.91e-05	7.92e-04	2.66e-04	4.28e-03
	Rank	8	1	7	6	3	5	4	2
<i>f</i> ₆	Mean	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Best	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<i>P</i> values	NA	NA	NA	NA	NA	NA	NA	NA
	Rank	#	#	#	#	#	#	#	#
<i>f</i> ₇	Mean	0.00	2.528e-14	25.64	28.91	1.347	27.66	2.44e-14	9.09e-05
	Best	0.00	7.105e-15	14.92	7.626	1.41e-15	12.58	0	4.87e-14
	SD	0.00	1.376e-14	5.393	14.60	2.699	7.697	5.979e-14	1.25e-04
	<i>P</i> values	NA	4.67e-14	3.01e-28	2.84e-29	3.18e-27	4.72e-28	4.62e-16	2.15e-20
	Rank	NA	NA	NA	NA	NA	NA	NA	NA

Table 3 continued

Function	Performance Metrics									
	DPG-PSO	ALCPSO	GPSO	FIPS	HPSO-TVAC	DMS-PSO	CLPSO	OLPSO		
f_8	Rank	1	2	6	8	5	7	3	4	
	Mean	8.88e-16	1.898e-14	1.245e-14	7.7e-15	8.22e-11	9.86e-15	2.964e-14	4.06e-07	
	Best	8.88e-16	7.012e-15	8.127e-15	4.1e-15	1.13e-13	7.44e-15	1.436e-14	1.25e-09	
	SD	0.00	2.453e-15	2.38e-15	5.9e-16	3.56e-10	1.41e-15	4.891e-15	7.68e-08	
	<i>P</i> values	NA	1.93e-03	7.39e-03	1.35e-02	2.29e-07	3.82e-02	1.63e-03	1.28e-09	
f_9	Rank	1	5	4	2	7	3	6	8	
	Mean	0.00	1.166e-02	1.759e-04	2.52e-04	1.967e-02	1.014e-02	1.847e-14	9.27e-09	
	Best	0.00	0.00	0.00	0.00	2.13e-15	0.00	0.00	0.00	
	SD	0.00	1.417e-02	1.44e-02	2.63e-04	1.009e-02	1.977e-02	8.168e-14	8.41e-09	
	<i>P</i> values	NA	3.75e-19	2.62e-18	3.85e-16	7.41e-10	3.88e-24	5.36e-12	6.39e-10	
f_{10}	Rank	1	6	8	4	7	5	2	3	
	Mean	0.00	2.458e-11	12.54	32.83	1.474	30.17	2.05e-01	8.49	
	Best	0.00	0.00	2.46	21.14	0.88e-14	18.63	0.00	1.56	
	SD	0.00	6.144e-11	15.411	5.841	2.854	6.098	3.649e-01	10.14	
	<i>P</i> values	NA	3.92e-10	5.30e-28	3.14e-27	7.28e-10	4.74e-24	5.12e-13	9.36e-26	
f_{11}	Rank	1	2	6	8	4	7	3	5	
	Mean	0.00	4.01e+03	4.88e+03	4.55e+03	5.47e+03	3.78e+03	4.27e+03	86.84	
	Best	0.00	2.15e+02	5.98e+02	5.41e+02	8.89e+02	3.43e+02	3.89e+02	12.45	
	SD	0.00	2.54e+02	6.72e+03	9.83e+02	7.96e+02	5.45e+02	3.88e+02	10.65	
	<i>P</i> values	NA	4.76e-20	3.56e-19	6.33e-21	4.72e-23	4.78e-27	1.67e-20	3.29e-08	
f_{12}	Rank	1	4	7	6	8	3	5	2	
	Mean	0.00	4.94e-07	59.84	1.2e2	52.0	42.35	4.73	5.40e-05	
	Best	0.00	8.17e-15	0.96	44.54	4.92e-01	9.96e-03	4.52e-03	3.57e-03	
	SD	0.00	5.78e-09	15.98	14.35	12.76	9.47	3.92	6.71e-05	
	<i>P</i> values	NA	3.27e-23	1.63e-20	5.18e-24	4.17e-25	7.93e-21	3.82e-22	5.80e-10	
f_{13}	Rank	1	2	7	8	6	5	4	3	
	Mean	0.00	5.86e-26	1.93	3.23e-07	9.67	2.44e-14	5.12e-05	7.42e-05	
	Best	0.00	9.45e-45	0.45	5.96e-09	1.68	8.04e-21	7.96e-07	4.55e-06	

Table 3 continued

Function	Performance Metrics									
	DPG-PSO	ALCPSO	GPSO	FIPS	HPSO-TVAC	DMS-PSO	CLPSO	OLPSO		
SD	0.00	4.67e-39	1.06	1.49e-07	2.07	1.96e-14	6.16e-05	7.44e-05		
	NA	3.55e-07	7.36e-18	2.68e-10	3.04e-20	1.66e-08	5.19e-15	2.86e-14		
	1	2	7	4	8	3	5	6		
f_{14}	1.4e+6	8.15e+5	1.84e+6	5.00e+6	5.84e+5	6.12e+6	1.53e+7	0.148e+6		
	1.1e+6	3.88e+5	4.4e+5	2.43e+6	1.85e+5	2.64e+6	7.41e+6	2.148e+5		
	2.7e+5	5.67e+5	7.87e+5	1.65e+6	3.01e+5	2.46e+6	4.60e+6	4.987e+5		
P values	3.61e-07	2.55e-05	1.83e-04	2.64e-10	NA	3.66e-08	2.81e-10	4.82e-08		
	4	2	5	6	1	7	8	3		
	Mean	26.34	22.216	67.17	41.00	50.897	15.68	404.16		
Best	22.31	4.85e-04	2.00e-01	22.10	1.745e-02	1.793e-03	4.79e-01	24.78		
	SD	1.02	40.69	54.756	20.287	70.625	20.594	13.84		
	P values	4.72e-04	5.36e-05	4.72e-10	2.88e-10	3.09e-06	2.45e-08	NA	3.54e-20	
Rank	3	2	6	7	4	5	1	8		
	Mean	23.234	23.444	29.597	24.735	22.408	25.872	25.463		
	Best	20.234	16.114	21.974	17.398	24.792	21.069	21.543		
SD	3.45	3.618	3.083	3.364	2.648	1.263	1.667	3.345		
	P values	3.51e-03	1.72e-02	4.27e-06	5.22e-03	2.84e-07	7.21e-03	2.66e-03		
	Rank	2	3	7	4	8	1	5		
f_{17}	Mean	112.23	111.633	121.84	178.32	54.040	100.753	165.33		
	Best	88.34	62.682	56.713	154.42	221.87	75.879	43.434		
	SD	21.434	28.375	55.083	11.488	52.897	8.419	26.43		
P values	1.29e-07	3.92e-07	1.25e-08	4.02e-08	5.83e-09	NA	3.25e-04	1.66e-06		
	Rank	4	3	5	7	1	2	6		
	Mean	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
Best	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
	SD	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
	P values	NA	NA	NA	NA	NA	NA	NA		
Rank	#	#	#	#	#	#	#	#		
	1.87	2.47	5.45	5.3	5.28	4.26	4.36	4.95		
	Average rank									

Best results have been marked in boldface

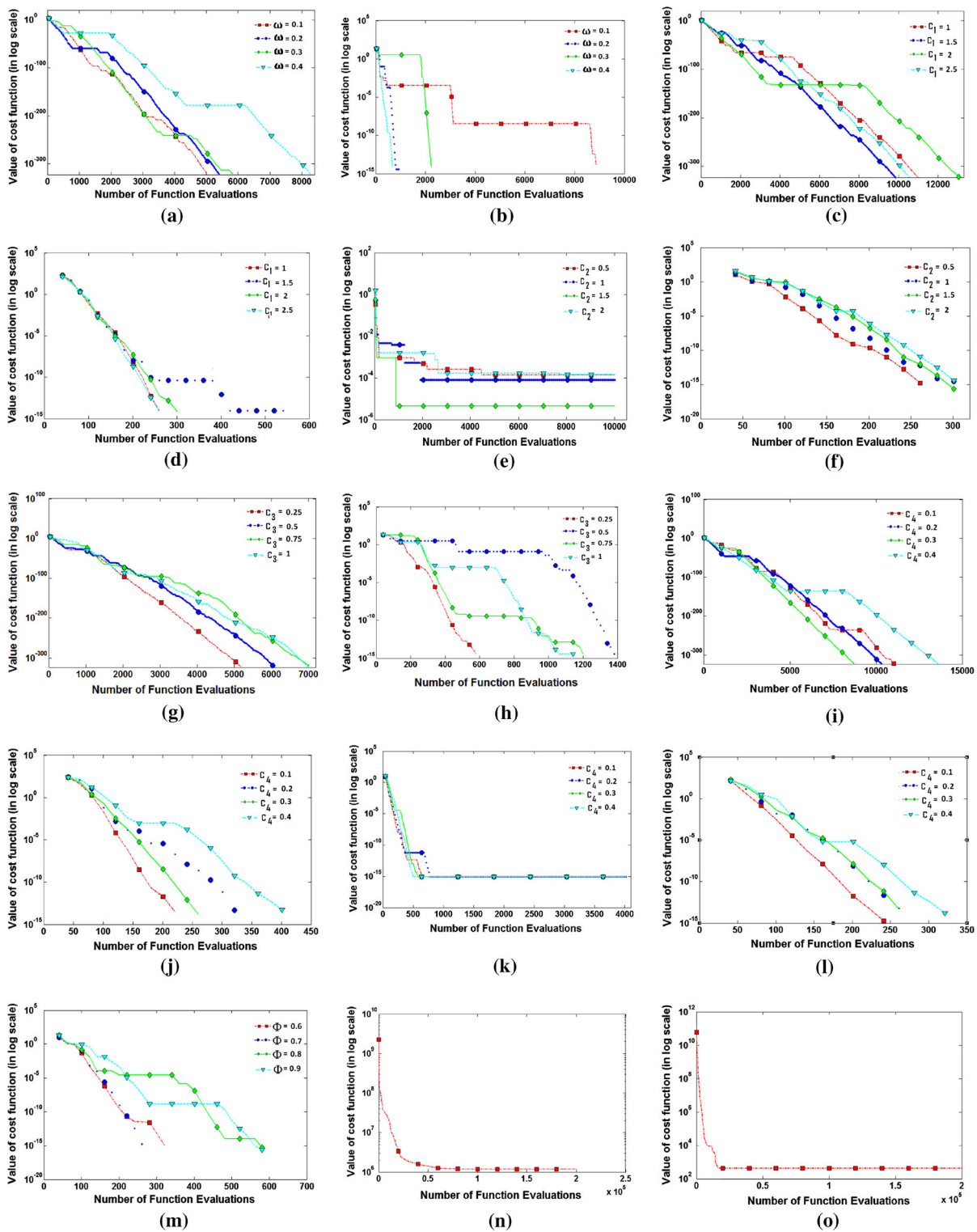


Fig. 6 Convergence characteristics os DPG-PSO for various parameter settings. **a** Varying ω_1 on sphere function f_1 **b** Varying ω_1 on f_{12} , **c** varying c_1 on Schwefel's problem 2.22 f_2 , **d** varying c_1 on Rastrigin function f_7 , **e** varying c_2 on unimodal Schwefel's problem 2.22 f_2 , **f** varying c_2 on Griewank function f_9 , **g** varying c_3 on Quadric function f_4 , **h** varying c_3 on rotated Ackley function f_{12} , **i** Varying c_4 on Schwefel's f_2 , **j** Varying c_4 on non-continuous Rastrigin f_{10} , **k** varying

c_4 on Ackley f_8 , **l** varying c_4 on rotated Rastrigin in f_{11} , **m** varying Φ on Griewank function f_9 , **n** convergence plot for shifted rotated high conditioned elliptic f_{14} , **o** convergence plot for shifted Rosenbrock's function f_{15} . **p** Convergence plot for shifted rotated Weierstrass f_{16} , **q** convergence plot for Shifted Rotated Rastrigin's f_{17} , **r** convergence plot for Shifted Sphere function f_{18}

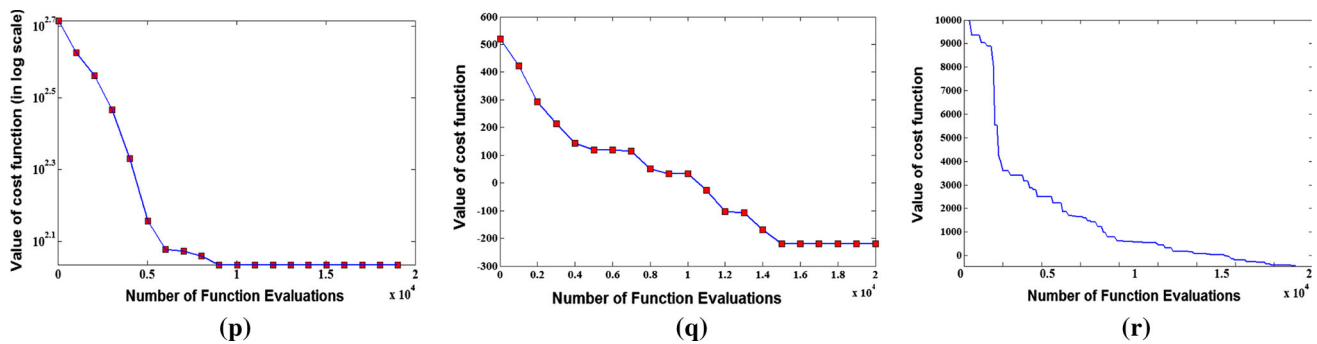


Fig. 6 continued

i, and k show that the DPG-PSO descends to the global optima quiet steeply. From Table 3, all the other techniques fail to optimize them decently, except f_6 where the mean and least error value obtained by all the algorithms are same and thus cannot be ranked. f_3 and f_5 are the only two functions where DPG-PSO stagnates and fails to reach the global minimum. But despite that, it manages to secure the 1st rank for f_3 when compared to other optimization techniques.

2) Multimodal functions:

Table 3 shows that for multimodal functions $f_7 - f_{10}$, DPG-PSO gives the least mean error as well as the mean best error among all the algorithms. Furthermore, the standard deviation over a period of 30 runs is 0 for this algorithm. Figures 6d, f, j, m show that the numerous local minima fail to interfere with DPG-PSO while other variants get stagnated in the local minima. DPG-PSO finds the global minima for all the test functions except f_8 , where it yields a mean error value of $8.88e-16$. However it still works better than other algorithms and secures the first rank for f_8 .

3) Shifted and rotated functions:

Table 3 indicates that, for rotated multimodal functions $f_{11} - f_{13}$ of table 3, DPG-PSO finds the global minima of the rotated functions successfully, shown in Fig. 6b, h, and l despite the complicacies. Additionally it also maintains zero standard deviations over a period of 30 runs. DPG-PSO performs more or less fairly for shifted functions $f_{14} - f_{18}$ when compared to other PSO variants. HPSO-TVAC and CLPSO obtain first ranks for functions f_{14} and f_{15} respectively, whereas DMS-PSO achieved first ranks for both functions f_{16} and f_{17} . DPG-PSO achieved ranks 4th, 3rd, 2nd, and 4th out of the 8 algorithms for these four functions, in respective order. Function f_{18} is included to show that the proposed algorithm performs at par with its peers for the shifted sphere function. The convergence plots of the shifted functions are shown in Fig. 6n, o, p, q, and r, respectively. For the shifted functions, and the shifted as well as rotated functions, none of the single PSO variants can perform

consistently good because of the complexity of the functions.

In terms of Wilcoxon's rank-sum test, this algorithms outperforms the all the other competing algorithms in all but three of the test functions, namely f_5 , f_{14} , and f_{15} . In f_{14} and f_{15} it still manages to secure a score of the order of 10^{-7} and 10^{-4} , respectively. In f_5 it falls behind only two algorithms to secure third rank with a score of the order of 10^{-5} .

In terms of average ranking, DPG-PSO stands as rank 1 with a score of 1.87 followed by ALC-PSO ranking 2.47, while other algorithms perform more or less in a similar manner obtaining average rank between 4 and 6. This demonstrates the efficiency of the proposed algorithm.

5 Parameter tuning

DPG-PSO contains 6 control parameters which control the swarm behavior of the particles. The inertia factor ω is varied between 0.1 and 0.5 to provide sufficient freedom to the particles to link and interact with one another. It is tested on unimodal sphere function and multimodal Ackley's function. It is found to have similar results in terms of convergence value and success rate. However, rate of convergence for $\omega = 0.1$ does not provide steep convergence for rotated Ackley function as it gets entrapped in a local minima. The acceleration coefficient c_1 is set between 1 and 2.5 to test its robustness. But it is found that performance of DPG-PSO is not sensitive to c_1 as test results on unimodal Schwefel's problem 2.22 and the multimodal Rastrigin's function show that it reaches the global minima for both the cases. The parameter c_2 is tested within the range of 0.5 to 2 and is found to be independent as found in convergence plots of Fig. 6e and f showing the successful convergence on unimodal Schwefel's P2.22 and rotated Griewank functions. Parameter c_3 is varied between (0, 1] as it is associated with an exponential term which tends to a very high value and thus the *Opposition* will overpower the *Governor* despite having low votes. It is found that DPG-PSO is also independent of c_3 as can be seen from Fig. 6g, h showing convergence plot of Unimodal

Fig. 7 Convergence plots of **a** unimodal Sphere function, **b** multimodal Rastrigin's function, and **c** rotated Griewank's function for democratic and undemocratic approaches used

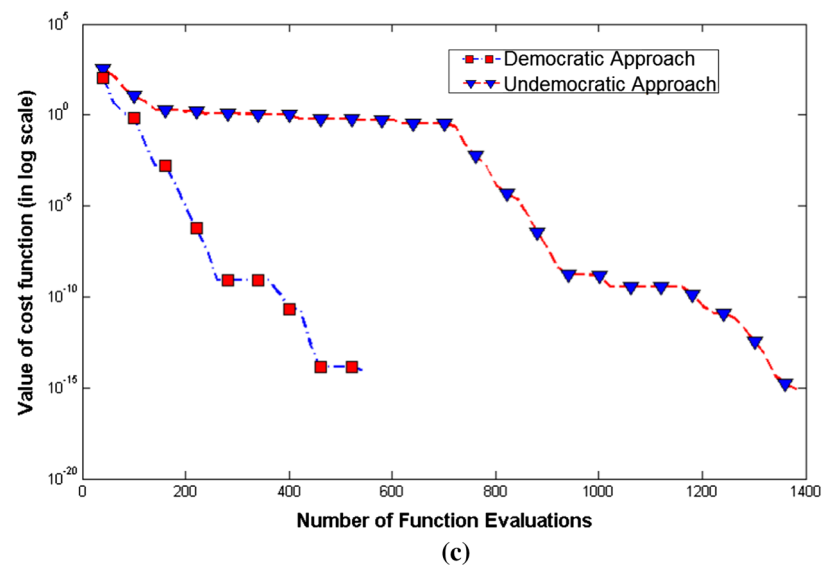
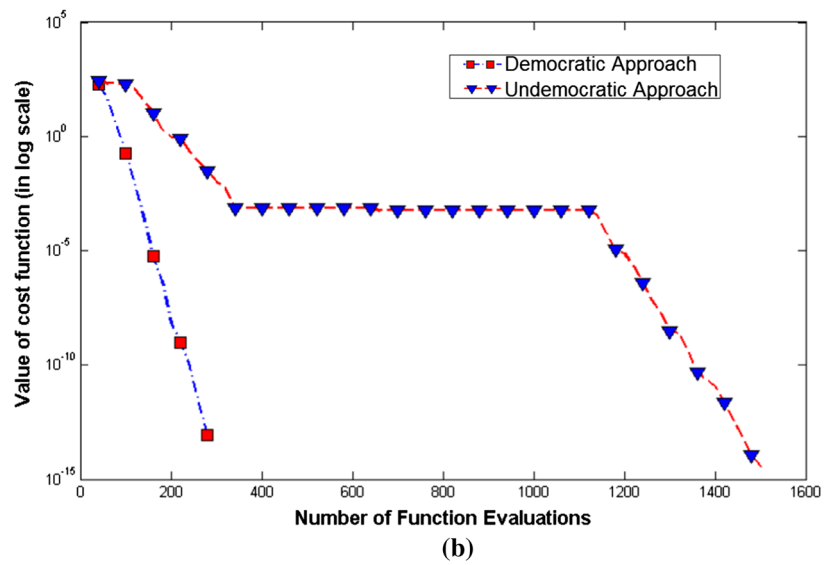
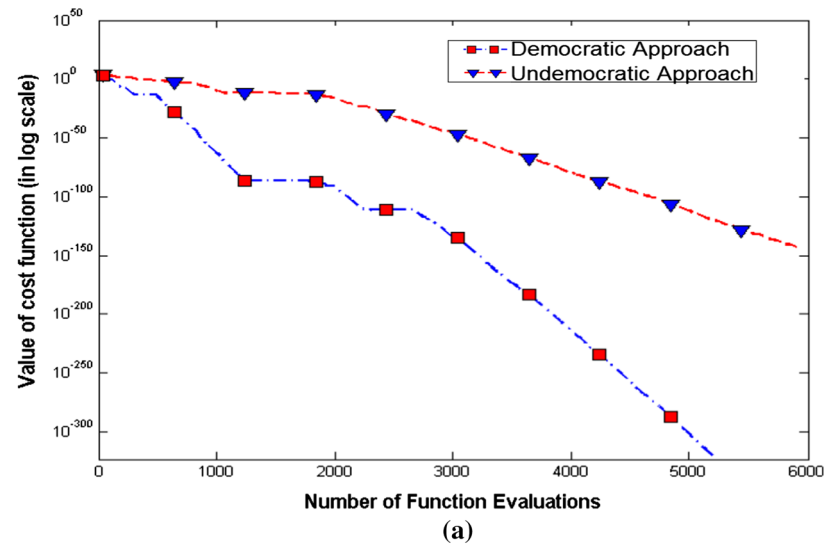


Table 4 Comparison of shifted functions over a period of 30 runs and 2,00,000 FEs, showing mean Error, minimum error, standard deviation, and *P* values from the Wilcoxon's rank-sum test

Shifted functions	Optima	Performance metrics	GPSO	DPG-PSO
f_2	$o = -10+20* \mathit{rand}(1, D)$	Mean error	0.0189	9.82e-126
		Least error	0.0166	2.406e-138
		Standard deviation	0.0016	1.768e-137
		<i>P</i> value	2.54e-18	NA
f_3	$o = -1.28+2.56* \mathit{rand}(1, D)$	Mean error	0.1542	2.81e-04
		Least error	0.1225	9.63e-05
		Standard deviation	0.0224	4.12e-05
		<i>P</i> value	1.72e-07	NA
f_4	$o = -100+200* \mathit{rand}(1, D)$	Mean error	8.1178e+03	6.532e-198
		Least error	7.1946e+03	2.765e-218
		Standard deviation	652.7402	4.641e-206
		<i>P</i> value	3.15e-20	NA
f_6	$o = -100+200* \mathit{rand}(1, D)$	Mean error	0.5000	0.00
		Least error	0.00	0.00
		Standard deviation	0.3536	0.00
		<i>P</i> value	1.66e-22	NA
f_8	$o = -32+64* \mathit{rand}(1, D)$	Mean error	10.7258	2.341e-14
		Least error	1.3476	2.341e-14
		Standard deviation	6.6314	0.00
		<i>P</i> values	3.84e-10	NA
f_9	$o = -600+1200* \mathit{rand}(1, D)$	Mean error	0.0057	1.34e-139
		Least error	5.9412e-04	2.94e-158
		Standard deviation	0.0036	8.88e-149
		<i>P</i> values	7.29e-20	NA

$\mathit{rand}(1, D)$ denotes a vector of D uniformly generated random numbers in $(0, 1)$

Quadric function and rotated Ackley function. Parameter c_4 is tuned in $(0.4, 1)$. It is found that the parameters do not have a significant influence on the convergence of the unimodal Schwefel's problem 2.22, multimodal non-continuous Rastrigin, multimodal Ackley, and rotated Rastrigin functions in Fig. 6i–l, respectively. Parameter Φ is varied from 0.6 to 0.9 to show the robustness. DPG-PSO is found to perform more or less in a similar manner, as can be seen in Fig. 6m for the convergence plot of Griewank function, even if is varied between the said limit.

6 Undemocratic approach

In this section, we investigate the performance of DPG-PSO when the *Opposition* is absent and there is no democracy. Hence, there will be a single *Leader* who is going to lead the entire population. So there will be no voting scheme. Only the Peer Group Topology will be present. This proves the efficiency of the democracy system. Three functions, one from each group of Unimodal, Multimodal, and Rotated and Shifted Functions are analyzed for their convergence plots with 6000 Function Evaluation as the ending criteria. All

the parameter values remain same. In Fig. 7a, it can be seen that the convergence plot of Sphere function attains its minima 0 within 5200 Function Evaluations for the proposed algorithm, but for the undemocratic approach, the convergence plot reaches a minima of around 10^{-150} and also the convergence rate is very slow. Further it can be seen from the convergence plots for multimodal Rastrigin function and rotated Griewank functions in Fig. 7b, c, respectively, that the scenario is pretty much same for all the cases as the convergence rate is slower for the undemocratic approach than the proposed algorithm. Also the undemocratic approach gets entangled to the local minima and remains stagnant for some Function Evaluations before proceeding again. This results in unnecessary consumption of time and Function Evaluation. Thus the democratic approach is far superior than the undemocratic approach.

7 The origin-seeking bias and DPG-PSO

Conventional PSO algorithms are often biased towards the center of the search space (Monson and Seppi 2005; Spears et al. 2010). If the search bounds are symmetrical (like

$\forall d, -L \leq x^d \leq L$, L being a constant), the center coincides with the origin (a vector of zeroes) and this is indeed the situation with all the 18 benchmarks reported in Table 2. If the corresponding function has its global optimum at origin, it may become easier for a PSO to detect the optimum. However, as has been shown in (Monson and Seppi 2005), if the functions have a center offset, i.e., global optimum is shifted to a location other than the origin of the feasible search volume, PSO can become fairly inefficient.

In this section, we demonstrate that the proposed DPG-PSO algorithm is not biased towards the global optimum at the absolute origin. For this we report the comparative results between DPG-PSO and the usual PSO algorithm (GPSO in Table 1) on the shifted versions of 6 functions (f_2, f_3, f_4, f_6, f_8 and f_9) from Table 2. Out of the 18 benchmarks shown in Table 2, these 6 functions were tested in their original forms (with global optimum lying at the absolute origin of the search space) in Table 3. The global optimum of each function is shifted by a random amount as indicated in Table 4. In this table, θ is the array containing the shift of each of the dimensions from the origin. The proposed algorithm is run 30 times and is compared with the basic variant of PSO. We report the mean error, least error, and standard deviation of the errors in Table 4. The same table also shows the P values obtained from Wilcoxon's rank-sum test between DPG-PSO and GPSO. A close look through Table 4 reveals that DPG-PSO yields statistically significantly better results than GPSO on all the instances of the shifted functions. The performance of GPSO on these shifted variants deteriorates considerably as compared to its performance on their original counterparts as can be seen from Table 3. These results indicate the absence of origin-seeking bias in DPG-PSO on the standard benchmarking instances.

8 Conclusion

In this paper, a new PSO variant called DPG-PSO has been proposed employing the concepts of democracy form of governance and social interaction of peer group. The new strategy of using two leaders gives freedom to the particles to choose their leader and breaks from the orthodox system of following a single leader. The proposed Peer-Based Topology enables the particles to learn more from its neighbors. These emulations of human behavior and style of governance helps DPG-PSO to efficiently surmount the problem of premature convergence. Experimental results on standard benchmark functions along with competitive algorithms, we can safely infer that DPG-PSO introduces a strategy to make PSO perform better and provide more accurate result with faster rate of convergence. It is also demonstrated that the proposed algorithm does not possess origin-seeking bias of the conventional PSO algorithm as it can keep up its good

performance on functions with randomly shifted optima. It is also not much sensitive to the control parameters and hence appears to be more robust.

For future research, the concept of democracy and peer group can be extended to other evolutionary computational algorithms and further studied. The democracy-based search strategies may also be extended to the multiobjective optimization scenarios.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Andrews PS (2006) An investigation into mutation operators for particle swarm optimization. In: Proc. IEEE Congr. Evol. Comput., pp 1044–1051
- Angeline PJ (1998) Using selection to improve particle swarm optimization. In: Proc. IEEE Congr. Evol. Comput., pp 84–89
- Banks A, Vincent J, Anyakoha C (2007) A review of particle swarm optimization. Part I: background and development. Nat Comput 6(4):467–484
- Blackwell TM, Bentley PJ (2002) Don't push me! Collision-avoiding swarms. Proc. IEEE Congr. Evol. Comput, Honolulu, HI, pp 1691–1696
- Bratto D, Kennedy J (2007) Defining a standard for particle swarm optimization. In: IEEE swarm intelligence symposium
- Chen WN et al. (2013) Particle Swarm Optimization with an Aging Leader and Challengers. In: IEEE Transactions on evolutionary computation 17(2):241–258
- Chen WN et al. (2012) Particle swarm optimization with an aging leader and challengers. IEEE Trans. on Evolutionary Computation (**In Press**)
- Chen YP, Peng WC, Jian MC (2007) Particle swarm optimization with recombination and dynamic linkage discovery. IEEE Trans Syst Man Cybern B 37(6):1460–1470
- Cho H, Kim D, Olivera F, Guikema S (2011) An enhanced Speciation in particle swarm optimization for multi-modal problems. Eur J Oper Res 213:15–23
- Clerc M (2008) Particle Swarm Optimization, ISTE Publications
- 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
- Das S, Konar A, Chakraborty UK (2005) Improving particle swarm optimization with differentially perturbed velocity. Proc. Genet. Evol. Comput. Conf. (GECCO), pp 177–184
- Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. IEEE Trans Evol Comput 13(3):526–553
- Dehuri S, Ghosh A, Mall R (2006) Particles with age for data clustering. 9-th Intl. Conf. on Inf. Tech. (ICIT), pp 221–224
- Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proc. 6th Int. Symp. Micromach. Human Sci, pp 39–43
- Eberhart RC, Shi Y (2001) Tracking and optimizing dynamic systems with particle swarms. Proc. IEEE Congr. Evol. Comput, Seoul, Korea, pp 94–97
- Eberhart RC, Shi Y (2004) Guest editorial. IEEE Trans. Evol. Comput.—Special Issue Particle Swarm Optimization 8(3):201–203
- Engelbrecht AP (2006) Fundamentals of Computational Swarm Intelligence. Wiley, USA

- Fan S-KS, Zahara E (2007) A hybrid simplex search and particle swarm optimization for unconstrained optimization. *Eur J Oper Res* 181:527–548
- Fishkin J (2005) Experimenting with a democratic ideal: deliberative polling and public opinion. In: *ActaPolitica*. pp 284–298
- Higashi N, Iba H (2003) Particle swarm optimization with Gaussian mutation. In: *Proc. IEEE Swarm Intell. Symp.*, pp 72–79
- Hu X, Eberhart RC (2002) Multiobjective optimization using dynamic neighborhood particle swarm optimization. In: *Proc. Congr. Evol. Comput*, Honolulu, HI
- Janson S, Middendorf M (2005) A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Trans Syst Man Cybern Part B Cybernetics*. 35(6)
- Kaveh A (2014) *Advances in meta-heuristic algorithm for optimal design of structures*. Springer, Switzerland
- Kaveh A, Zolghadr A (2013) Democratic PSO with truss layout and size optimization with frequency constraints. *Comput Struct* 130:10–21
- Kennedy J (1997) The particle swarm social adaptation of knowledge. In: *Proc. IEEE Int. Conf. Evol. Comput*, Indianapolis, IN, pp 303–308
- Kennedy J (1999) Small worlds and mega-minds: effects of neighbourhood topology on Particle Swarm performance. In: *Proc. Congr. Evol. Comput.*, pp 1931–1938
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4., pp 1942–1948
- Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: *Proc. IEEE Congr. Evol. Comput.*, vol 2. pp 1671–1676
- Kennedy J, Mendes R (2006) Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Trans Syst Man Cybern C Appl Rev* 36(4):515–519
- 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, Suganthan PN (2005) Dynamic multiswarm particle swarm optimizer. In: *Proc. Swarm Intell. Symp.*, pp 124–129
- Liao CC, Zhao XL, Xu JZ (2012) Blade layers optimization of wind turbines using FAST and improved PSO Algorithm. *Renew Energy* 42:227–233
- Lim WH, Isa NAM (2014) An adaptive two-layer particle swarm optimization with elitist learning strategy. *Inf Sci* 273:49–72
- Lovbjerg M, Krink T (2002) Extending Particle Swarm optimizers with self-organized criticality. In: *Proc. Congr. Evol. Comput*, Honolulu, HI, pp 1588–1593
- Lovbjerg M, Rasmussen TK, Krink T (2001) Hybrid particle swarm optimizer with breeding and subpopulations. In: *Proc. Genet. Evol. Comput. Conf.*, pp 469–476
- Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. In: *IEEE Trans. Evol. Comput.* 8:204–210
- Miranda V, Fonseca N (2002) New evolutionary Particle Swarm algorithm (EPSO) applied to voltage/VAR control. In: *Proc. 14th Power Syst. Comput. Conf.*, Seville, Spain
- Monson CK, Seppi KD (2005) Exposing origin-seeking bias in PSO”. *Genetic and Evolutionary Computation Conference (GECCO '05)*, pp 241–248
- Nasir M, Das S, Maity D, Sengupta S, Halder U, Suganthan PN (2012) A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Inf Sci* 209:16–36
- Omran MGH, Engelbrecht AP, Salman A (2006) Using the ring neighborhood topology with self-adaptive differential evolution. *Adv Nat Comput* 4221:976–979
- Parsopoulos KE, Vrahatis MN (2004) On the computation of all global minimizers through particle swarm optimization. *IEEE Trans Evol Comput* 8(3):211–224
- Poli R (2008) Analysis of the publications on the applications of particle swarm optimisation. *J Artif Evol Appl* 10. doi:10.1155/2008/685175
- Qu B-Y, Suganthan PN, Das S (2012) A distance-based locally informed particle swarm model for multi-modal optimization. *IEEE Trans Evol Comput*. doi:10.1109/TEVC.2012.2203138
- Quande Q, Li L, Rongjun L (2010) A novel PSO with piecewise-varied inertia weight. In: *2010 2nd IEEE International Conference on Information and Financial Engineering (ICIFE)*, pp 503–506
- Ratnaweera A, Halgamuge S, Waston H (2004) Self-organizing hierarchical particle optimizer with time-varying acceleration coefficients. *IEEE Trans Evol Comput* 8(3):240–255
- Shayeghi H, Mahdavi M, Bagheri A (2010) Discrete PSO algorithm based optimization of transmission lines loading in TNEP problem. *Energy Conv Manag* 51(1):112–121
- Shelokar PS, Siarry P, Jayaraman VK, Kulkarni BD (2007) Particle swarm and ant colony algorithms hybridized for improving continuous optimization. *Appl Math Comput* 188(1):129–142
- Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: *Proc. IEEE Congr. Evol. Comput*. pp 69–73
- Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: *Proc. IEEE Congr. Evol. Comput.*, pp 1945–1950
- Shi Y, Eberhart RC (2001) Fuzzy adaptive particle swarm optimization. In: *Proc. IEEE Congr. Evol. Comput.*, vol 1, pp 101–106
- Spears WM, Green DT, Spears DF (2010) Biases in particle swarm optimization. *Int J Swarm Intell Res* 1(2):34–57
- Suganthan PN (1999) Particle swarm optimizer with neighborhood operator. In: *Proc. Congr. Evol. Comput*, Washington, DC, pp 1958–1962
- Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, Nanyang Technological University
- Tillett J, Mao TM, Sahin F, Rao R (2005) Darwinian particle swarm optimization. In: *Proc. Indian Int. Conf. Artif. Intell.*, pp 1474–1487
- Valle YD, Venayagamoorthy GK, Mohagheghi S, Hernandez JC, Harley RG (2008) Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Trans Evol Comput* 12(2):171–195
- Van den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. *IEEE Trans Evol Comput* 8(3):225–239
- Venter G, Sobieski JS (2003) “Particle swarm optimization”. *AIAA J* 41(8):1583–1589
- Wang L, Yang B, Chen Y (2014) Improving particle swarm optimization using multi-layer searching strategy. In: *Information Sciences*, vol 274, pp 70–94
- Xin B, Chen J, Zhang J, Fang H, Peng ZH (2012) Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. *IEEE Trans SMC Part C* 42(5):744–767
- Zhan ZH, Zhang J, Li Y, Shi Y (2011) Orthogonal learning particle swarm optimization. *IEEE Trans Evol Comput* 15(6):832–847
- Zhang WJ, Xie XF (2003) DEPSO: hybrid particle swarm with differential evolution operator. In: *Proc. IEEE Int. Conf. Syst. Man Cybern.*, pp 3816–3821