Applied Mathematics and Computation 219 (2013) 5817–5836

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

MPSO: Median-oriented Particle Swarm Optimization

Zahra Beheshti, Siti Mariyam Hj. Shamsuddin *, Shafaatunnur Hasan

Soft Computing Research Group, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Skudai, 81310 Johor, Malaysia

ARTICLE INFO

Keywords:
Optimization
Particle Swarm Optimization
Median-oriented Particle Swarm Optimization
Global optimum
Local optimum

ABSTRACT

Particle Swarm Optimization (PSO) is a bio-inspired optimization algorithm which has been empirically demonstrated to perform well on many optimization problems. However, it has two main weaknesses which have restricted the wider applications of PSO. The algorithm can easily get trapped in the local optima and has slow convergence speed. Therefore, improvement and/or elimination of these disadvantages are the most important objective in PSO research. In this paper, we propose Median-oriented Particle Swarm Optimization (MPSO) to carry out a global search over entire search space with accelerating convergence speed and avoiding local optima. The median position of particles and the worst and median fitness values of the swarm are incorporated in the standard PSO to achieve the mentioned goals. The proposed algorithm is evaluated on 20 unimodal, multimodal, rotated and shifted high-dimensional benchmark functions and the results are compared with some well-known PSO algorithms in the literature. The results show that MPSO substantially enhances the performance of the PSO paradigm in terms of convergence speed and finds global or good near-global optimal in the functions.

1. Introduction

In the past decades, the design of nature-inspired meta-heuristic algorithms has dramatically increased to deal with different optimization problems. One of the most famous algorithms is Particle Swarm Optimization (PSO) [1,2] inspired by the social behavior of bird flocking, fish schooling and swarm theory. Due to simple concepts, easy to implement and having few parameters, the original or modified PSO has been extensively applied in solving various optimization problems [3–7]. However, PSO suffers from two main drawbacks of premature convergence rate and trapping into local minima when solving complex multimodal problems. Therefore, variant PSO algorithms have been proposed to overcome the weaknesses. To mention a few, Local ring topology PSO (LPSO) [8], Von Neumann topological structure PSO (VPSO) [9], Fully Informed Particle Swarm (FIPS) [10], self-organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients (HPSO-TVAC) [11], Dynamic Multi-Swarm PSO (DMS-PSO) [12] and Comprehensive Learning PSO (CLPSO) [13] are several samples. Also, some other techniques were applied to improve the performance of PSO which combine operators with original PSO such as mutation [14], crossover [15,16], selection [17], elitist learning strategy [18], orthogonal learning strategy [19,20] and chaos [21].

Nevertheless, there is no specific algorithm to achieve the best solution for all optimization problems [22]. In other words, some of them provide better results for particular problems than others. Therefore, performance enhancement of previous optimization algorithms or proposing new ones would seem necessary. In this paper, Median-oriented Particle Swarm Optimization (MPSO) is proposed to improve PSO algorithm in both terms of accelerating convergence rate and avoiding local

* Corresponding author.
E-mail addresses: bzahra2@live.utm.my (Z. Beheshti), mariyam@utm.my (Siti Mariyam Hj. Shamsuddin), shafaatunnur@gmail.com (S. Hasan).

http://dx.doi.org/10.1016/j.amc.2012.12.013

optima. Simulations and comparisons based on unimodal, multimodal, rotated and shifted benchmark functions indicate that the proposed algorithm is more effective than other compared algorithms.

The structure of this paper is organized as follows. In Section 2, a brief review of related works is presented. The details of MPSO are explained in Section 3. In Section 4, the experimental results on some well-known benchmark functions are shown. Finally, conclusions are demonstrated in Section 5.

2. Related works

PSO is a global optimization algorithm made up of simple particles interacting locally with each other and with their environment. Local and random interactions among the particles are led to an intelligent global behavior for finding good solutions.

Each particle is described by a group of vectors denoted as $(\vec{X}_i, \vec{V}_i, \vec{P}_i)$ in a d-dimensional search space, where $\vec{X}_i$ and $\vec{V}_i$ are the position and velocity of the ith particle defined as:

$$\vec{X}_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \quad \text{for } i = 1, 2, \ldots, N. \tag{1}$$

$$\vec{V}_i = (v_{i1}, v_{i2}, \ldots, v_{id}) \quad \text{for } i = 1, 2, \ldots, N. \tag{2}$$

$\vec{P}_i$ is the personal best position found by the ith particle:

$$\vec{P}_i = (p_{i1}, p_{i2}, \ldots, p_{id}) \quad \text{for } i = 1, 2, \ldots, N. \tag{3}$$

Also, the best position achieved by the entire swarm population ($\vec{P}_g$) is computed to update the particle velocity:

$$\vec{P}_g = (p_{g1}, p_{g2}, \ldots, p_{gd}). \tag{4}$$

From $\vec{P}_i$ and $\vec{P}_g$, the next velocity and position of ith particle are acquired by Eqs. (5) and (6):

$$v_{id}(t+1) = w(t) \times v_{id}(t) + C_1 \times rand \times (p_{id}(t) - x_{id}(t)) + C_2 \times rand \times (p_{gd}(t) - x_{id}(t)), \tag{5}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \tag{6}$$

where $v_{id}(t+1)$ and $v_{id}(t)$ are the next and current velocity of ith particle respectively. w is inertia weight, C_1 and C_2 are acceleration coefficients, $rand$ is uniformly random number in the interval of [0,1] and N is the number of particles. $x_{id}(t+1)$ and $x_{id}(t)$ show the next and current position of ith particle. In Eq. (5), the second and the third term are called cognition and social term respectively. Also, $|v_{id}| < v_{\max}$ is and $v_{\max}$ is set to a constant based on the bounds of solution space by users. A larger value of w encourages a global exploration (searching new areas) while a smaller inertia weight facilitates a local exploitation [2].

In PSO algorithm, two models for choosing $\vec{P}_g$ are considered known as gbest (or global topology) and lbest (or local topology) models. In global model, the position of each particle is affected by the entire best-fitness of swarm in the search space whereas in the local model, each particle is influenced by the best-fitness particle chosen from its neighborhood. In this paper, the PSO algorithm which utilizes lbest model is called LPSO.

To improve the performance of PSO, many studies have been performed through parameter selecting [5,11,24], integration of its self-adaptation [6,23] and swarm topology. Kennedy and Mendes proposed a ring topological structure PSO (LPSO) [8] and a Von Neumann topological structure PSO (VPSO) [9] to avoid premature convergence in solving multimodal problems. In 2004, Mendes et al. suggested the Fully Informed Particle Swarm (FIPS) algorithm [10] which applied the information of the entire neighborhood to guide the particles. Dynamic Multi-Swarm PSO (DMS-PSO) [12] was introduced by Liang and Suganthan to enhance the topological structure in a dynamic way. The quadratic interpolation PSO (QIPSO) algorithm [16] introduced a quadratic crossover operator to improve numerical results. Ratnaweera et al. [11] proposed HPSO-TVAC algorithm which used linearly Time-Varying Acceleration Coefficients, where a larger C_1 and a smaller C_2 were set at the beginning and were gradually reversed during the search. In another research, an Adaptive Fuzzy Particle Swarm Optimization (AFPSO) [23] proposed which utilized fuzzy inferences to adaptively adjust acceleration coefficients instead of using fixed constants. Also, the quadratic crossover operator [16] used in the proposed AFPSO algorithm (AFPSO-QI) [23] to solve multimodal functions algorithm. Parsopoulos and Vrahatis [25] integrated exploration and exploitation to form a Unified Particle Swarm Optimization (UPSO). Moreover, Liang et al. [13] presented Comprehensive Learning Particle Swarm Optimization (CLPSO) in 2006. The algorithm, focused on avoiding the local optima by encouraging each particle to learn from other particles on different dimensions. An Adaptive Particle Swarm Optimization (APSO) [18] was proposed by Zhan et al. in 2009. The algorithm applied a real-time evolutionary state estimation procedure and an elitist learning strategy to accelerate convergence speed and avoid the local optima. In 2011, Zhan et al. [20] introduced Orthogonal Learning Particle Swarm Optimization algorithm (OLPSO). The OL strategy could guide particles to discover useful information from the personal best position and its neighborhood's best position in order to fly in better directions. In another study, Gao et al. [26] used PSO with chaotic opposition-based population initialization and stochastic search technique to solve complex multimodal problems. The algorithm, which was called CSPSO found new solutions in the neighborhoods of the previous best positions in order to escape from local optima in multimodal functions.

Although, the different type of PSO algorithms has been proposed so far, researches are still interested in improving the performance of the PSO in complex problems. With that regard, in the next section, the new algorithm named Median-oriented Particle Swarm Optimization (MPSO) is introduced.

3. MPSO – the proposed algorithm

In this section, an improved scheme of PSO called Median-oriented Particle Swarm Optimization (MPSO) is presented to enhance the performance of PSO. In PSO, if a particle falls into a local optimum, sometimes it cannot get rid of itself from the position. In other words, if $\vec{P}_g$ obtained by the population is a local optimum and the current position and the personal best position of particle i are in the local optimum, the second and third term of Eq. (5) tend toward zero, also w is linearly decreasing to near zero. Therefore, the next velocity of particle i tends toward zero and the next position in Eq. (6) cannot be updated. Thus, the particle remains in the local optimum. To solve the problem and to accelerate convergence speed of PSO, MPSO incorporates some terms into PSO equations.

In MPSO, each particle is shown by $(\vec{X}_i, \vec{V}_i, \vec{P}_i)$, in a d-dimensional search space. The particle position, $\vec{X}_i$, and velocity, $\vec{V}_i$, the personal best position, $\vec{P}_i$ and the best position explored by entire particles, $\vec{P}_g$, are defined as Eqs. (1)–(4). As originally proposed for PSO we consider two approaches of choosing $\vec{P}_g$ known as gbest and lbest models. The later model is called LMPSO in this paper. Each particle updates its velocity based on Eq. (7):

$$v_{id}(t+1) = v_{id}(t) + M_{id}(t) \tag{7}$$

where $v_{id}(t+1)$ and $v_{id}(t)$ are the next and current velocity respectively. Also, $M_{id}(t)$ represents the median-oriented acceleration which is defined as follows:

$$M_{id}(t) = a_i(t) \times [rand \times (p_{id}(t) - p_{md}(t) - x_{id}(t)) + rand \times (p_{gd}(t) - p_{md}(t) - x_{id}(t))], \tag{8}$$

where $rand$ is a random variable with uniform distribution between 0 and 1. $x_{id}(t)$ is the current position, $p_{id}(t)$ and $p_{gd}(t)$ are the personal best position of ith particle and the global best position explored so far by the population, also, $p_{md}(t)$ is the current median position of the swarm in the dth dimension. $a_i(t)$ is acceleration factor as:

$$A_i(t) = \frac{fit_i(t) - Maxfit(t)}{Medfit(t) - Maxfit(t)}, \tag{9}$$

$$a_i(t) = \frac{A_i(t)}{\sum_{j=1}^{N} A_j(t)}, \tag{10}$$

where $fit_i(t)$ represents the fitness value of the particle i, $Maxfit(t)$ and $Medfit(t)$ are the current maximum and median fitness values of swarm:

$$Maxfit(t) = \max(fit_j(t)) \quad \text{for } j = 1, 2, \ldots, N. \tag{11}$$

$$Medfit(t) = median(fit_j(t)) \quad \text{for } j = 1, 2, \ldots, N. \tag{12}$$

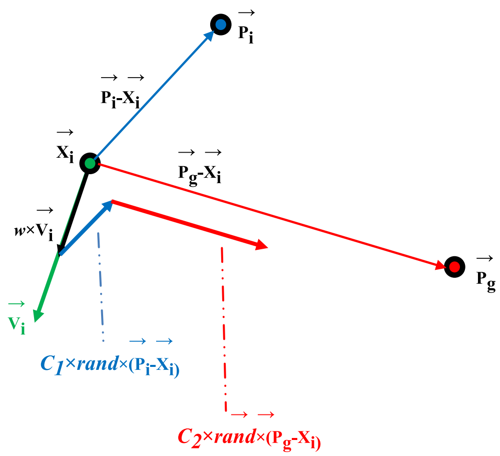

Fig. 1. Cognition and social terms in PSO.

In PSO, the cognition and social terms move a particle toward good solutions based on the particle experience and the best solution found by the swarm in the search space as shown in Fig. 1. Also, in classical or Newtonian's mechanics, the position vector of a particle subject to constant acceleration during the interval Δt is computed by [27]:

$$x_2 = x_1 + v_1 \cdot \Delta t + \frac{1}{2} \cdot a \cdot \Delta t^2, \tag{13}$$

where x_1 and x_2 are initial and final position, a and v_1 represent the particle's acceleration and velocity respectively.

Hence, these terms are applied for updating the next particle position in MPSO. In other words, the cognition and social terms in PSO is used as particle acceleration in MPSO to update the next particle position, $x_{id}(t+1)$, and Eq. (13) in MPSO is defined as follows:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) + \frac{1}{2} \times [rand \times (p_{id}(t) - x_{id}(t)) + rand \times (p_{gd}(t) - x_{id}(t))]. \tag{14}$$

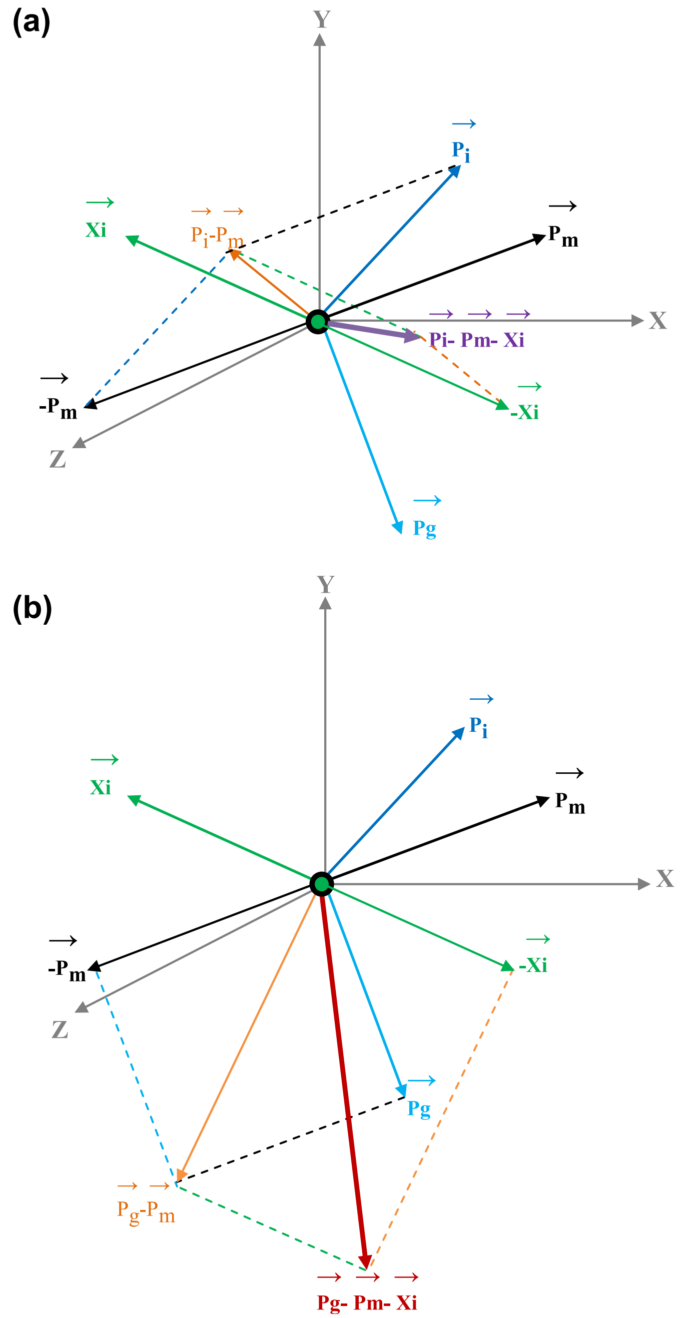

Fig. 2. Graphical representation: (a) $(\vec{P}_i - \vec{P}_m - \vec{X}_i)$ in $\vec{M}_i$, (b) $(\vec{P}_g - \vec{P}_m - \vec{X}_i)$ in $\vec{M}_i$ and (c) $\vec{M}_i$.

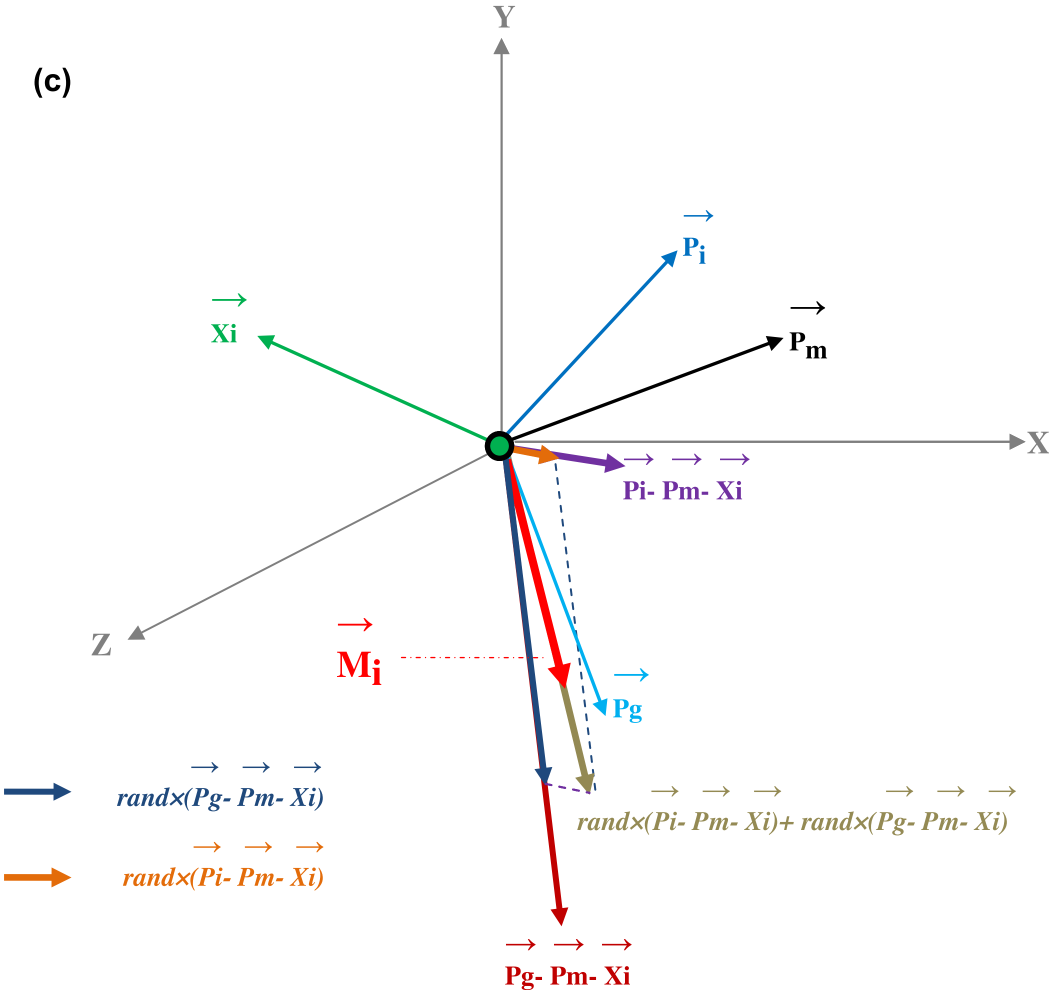

Fig. 2. (*continued*)

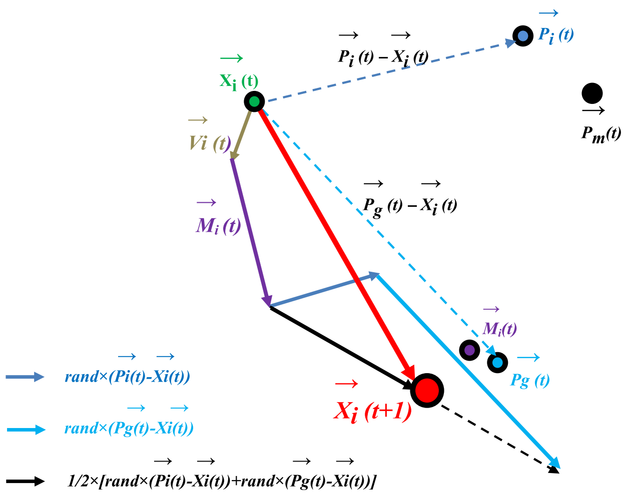

Fig. 3. Graphical representation of the next position of particle.

Figs. 2 and 3 demonstrate the graphical representation of $\vec{M}_i$ and $\vec{X}_i(t+1)$ respectively. The term of median-oriented acceleration plays a key role to increase the convergence speed and to escape from the local optima. If a particle is far from the solutions obtained so far by the swarm ($\vec{P}_g$, $\vec{P}_i$, $\vec{P}_m$), this term helps to move the particle toward the solutions. In PSO, w in Eq. (5) is set to a value at the beginning and usually is decreased linearly during the running algorithm in order to balance between the exploration and exploitation process. In MPSO, the median-oriented acceleration performs a proper tradeoff between the exploration and exploitation.

Figs. 4 and 5 illustrate the pseudo code and flowchart of MPSO. As shown in these figures, the velocities and positions of particles are randomly initialized and each particle is evaluated based on its fitness value. $\vec{P}_i$ is computed for each particle and the best position explored by the swarm is selected as $\vec{P}_g$. The next particles velocity and position will be acquired and these steps will be continued until stopping criterion is met. Finally, the best solution is returned by the algorithm.

Step 1: *Start.*

Step 2: *Initialize the velocities and positions of population randomly.*

Step 3: *Evaluate fitness values of particles.*

Step 4: *Update Pi if particle fitness value $_i$ < particle best fitness value $_i$, for i = 1, 2, . . ., N.*

Step 5: *Update Pg if particle fitness value $_i$ < global best fitness value, for i = 1, 2, . . ., N.*

Step 6: *Calculate $M_i(t)$ for i = 1, 2, . . ., N as Eq. (8).*

Step 7: *Update the next velocities of particles as Eq. (7).*

Step 8: *Update the next positions of particles as Eq. (14).*

Step 9: *Repeat steps 3 to 8 until the stop criterion is reached.*

Step 10: *Return the best solution.*

Fig. 4. MPSO pseudo code.

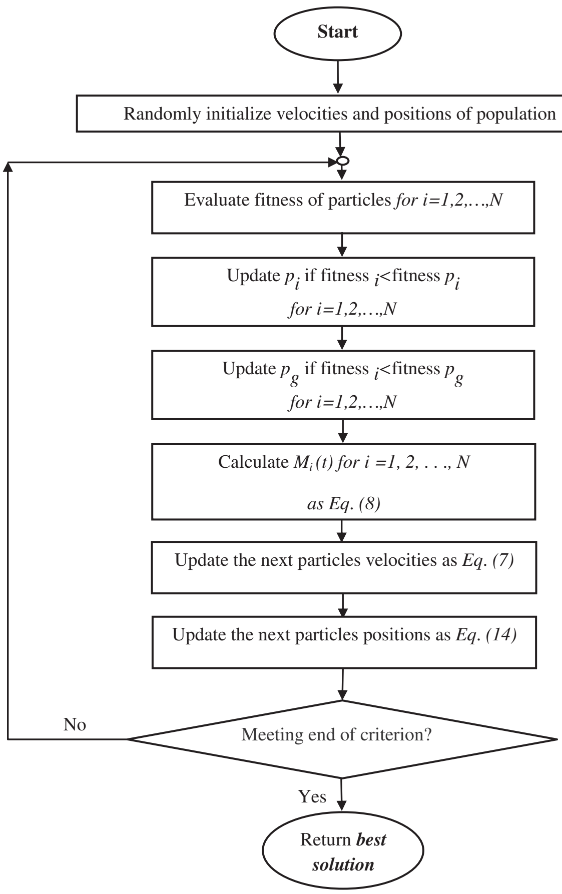

Fig. 5. MPSO flowchart.

4. Experimental results and discussion

To evaluate the performance of MPSO and LMPSO, 20 minimization benchmark functions are selected [13,28–30] as detailed in Section 4.1. The proposed algorithms are compared with PSO and LPSO and the results are given in Section 4.2. In addition, MPSO is compared with some well-known PSO algorithms in Section 4.3.

4.1. Benchmark functions

In this experimental study, 20 minimization functions including unimodal, multimodal, rotated, shifted and shifted rotated functions are used as detailed in the following section and Table 1.

Among the benchmarks, functions 1–6 are unimodal functions and functions 7–13 are in the class of multimodal functions. Functions 14–17 are rotated multimodal and unimodal functions. Function 18 is shifted unimodal and functions 19 and 20 are shifted rotated multimodal functions with $x = [x_1, x_2, \ldots, x_n]$ and the shifted global optimum $O = [o_1, o_2, \ldots, o_n]$. To obtain a rotated function, an orthogonal matrix M [31] is considered and the rotated variable $y = M \times x$ is computed. The variable y is used to evaluate the fitness value F. Therefore, the rotated function increases the function complexity and does not affect the shape of function. In unimodal functions, the convergence rate of the search algorithm is more interesting than the final results because other methods have been designed to optimize these kinds of functions. In multimodal functions, finding the optimum (or a good near-global optimum) solution is important. These functions are more difficult to optimize because the number of local optima increases exponentially with growing dimension. Therefore, the search algorithms should not be trapped into a local optimum and should be able to obtain good solutions.

In Table 1, *Range* and n are the feasible bound and the dimension of function respectively. F_{opt} is also the optimum value of function.

1. Sphere Model (unimodal function)

$$F_1(x) = \sum_{i=1}^{n} x_i^2$$

2. Schewefel' s Problem 2.22 (unimodal function)

$$F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$$

3. Schwefel's Problem 1.2 (unimodal function)

$$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$$

Table 1
Dimensions, ranges, and global optimum values of test functions used in the experiments.

Test function	Dimension (n)	$[\text{Range}]^n$	F_{opt}
$F_1(x)$	30/100/200	$[-100,100]^n$	0
$F_2(x)$	30/100/200	$[-10,10]^n$	0
$F_3(x)$	30/100/200	$[-100,100]^n$	0
$F_4(x)$	30/100/200	$[-100,100]^n$	0
$F_5(x)$	30/100/200	$[-100,100]^n$	0
$F_6(x)$	30/100/200	$[-1.28,1.28]^n$	0
$F_7(x)$	30/100/200	$[-5.12,5.12]^n$	0
$F_8(x)$	30/100/200	$[-5.12,5.12]^n$	0
$F_9(x)$	30/100/200	$[-32,32]^n$	0
$F_{10}(x)$	30/100/200	$[-600,600]^n$	0
$F_{11}(x)$	30/100/200	$[-0.5,0.5]^n$	0
$F_{12}(x)$	30/100/200	$[-50,50]^n$	0
$F_{13}(x)$	30/100/200	$[-1,1]^n$	$-0.1 \times n$
$F_{14}(x)$	30/100/200	$[-5.12,5.12]^n$	0
$F_{15}(x)$	30/100/200	$[-100,100]^n$	0
$F_{16}(x)$	30/100/200	$[-100,100]^n$	0
$F_{17}(x)$	30/100/200	$[-1.28,1.28]^n$	0
$F_{18}(x)$	30/100/200	$[-100,100]^n$	−450
$F_{19}(x)$	30/100/200	$[-32,32]^n$	−140
$F_{20}(x)$	30/100/200	$[-0.5,0.5]^n$	90

4. Schwefel's Problem 2.21 (unimodal function)

$$F_4(x) = \max_i\{|x_i|, 1 \leqslant i \leqslant n\}$$

5. Step Function (unimodal function)

$$F_5(x) = \sum_{i=1}^{n}(\lfloor x_i + 0.5 \rfloor)^2$$

6. Quartic Function, i.e. Noise (unimodal function)

$$F_6(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$$

7. Generalized Rastrigin's Function (multimodal function)

$$F_7(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$$

Table 2
Minimization results of unimodal and multimodal functions (iteration = 5000 and n = 30).

Functions		MPSO	PSO	LMPSO	LPSO
F_1	Avg. best solution	$\mathbf{1.67 \times 10^{-45}}$	9.92×10^{-33}	2.87×10^{-35}	5.80×10^{-13}
	SD	$\mathbf{2.80 \times 10^{-45}}$	2.36×10^{-32}	5.47×10^{-35}	7.44×10^{-13}
	Median Best solution	7.84×10^{-46}	1.18×10^{-33}	1.49×10^{-35}	2.34×10^{-13}
F_2	Avg. best solution	$\mathbf{9.49 \times 10^{-35}}$	9.00	3.12×10^{-19}	8.02×10^{-10}
	SD	$\mathbf{1.33 \times 10^{-34}}$	7.59	3.95×10^{-19}	6.32×10^{-10}
	Median Best solution	6.48×10^{-35}	10.00	2.37×10^{-19}	5.14×10^{-10}
F_3	Avg. best solution	$\mathbf{2.66 \times 10^{-122}}$	$1.06 \times 10^{+04}$	5.60×10^{-37}	$3.06 \times 10^{+03}$
	SD	$\mathbf{1.01 \times 10^{-121}}$	$6.82 \times 10^{+03}$	8.90×10^{-37}	$1.71 \times 10^{+03}$
	Median Best solution	2.09×10^{-123}	$1.00 \times 10^{+04}$	2.32×10^{-37}	$2.40 \times 10^{+03}$
F_4	Avg. best solution	$\mathbf{1.47 \times 10^{-46}}$	0.958	2.79×10^{-19}	10.51
	SD	$\mathbf{1.88 \times 10^{-46}}$	0.487	2.79×10^{-19}	1.89
	Median Best solution	8.50×10^{-47}	0.823	2.01×10^{-19}	10.69
F_5	Avg. best solution	**0.00**	**0.00**	**0.00**	**0.00**
	SD	**0.00**	**0.00**	**0.00**	**0.00**
	Median Best solution	0.00	0.00	0.00	0.00
F_6	Avg. best solution	$\mathbf{4.17 \times 10^{-06}}$	0.458	5.96×10^{-04}	3.41×10^{-02}
	SD	$\mathbf{4.46 \times 10^{-06}}$	1.24	1.73×10^{-04}	9.40×10^{-03}
	Median Best solution	3.34×10^{-06}	1.16×10^{-02}	5.79×10^{-04}	3.55×10^{-02}
F_7	Avg. best solution	0.00	60.97	9.26×10^{-09}	54.29
	SD	0.00	27.51	4.80×10^{-08}	14.96
	Median Best solution	0.00	59.27	0.00	53.73
F_8	Avg. best solution	0.00	57.97	87.73	62.12
	SD	0.00	23.31	28.50	19.04
	Median Best solution	0.00	54.50	82.78	61.41
F_9	Avg. best solution	$\mathbf{4.44 \times 10^{-15}}$	1.46×10^{-14}	$\mathbf{4.44 \times 10^{-15}}$	1.56×10^{-06}
	SD	**0.00**	4.25×10^{-15}	**0.00**	2.87×10^{-06}
	Median Best solution	4.44×10^{-15}	1.51×10^{-14}	4.44×10^{-15}	$5.36 \times 10-07$
F_{10}	Avg. best solution	0.00	$1.63 \times 10-02$	**0.00**	3.29×10^{-04}
	SD	0.00	1.64×10^{-02}	**0.00**	1.80×10^{-03}
	Median Best solution	0.00	1.11×10^{-02}	0.00	1.83×10^{-11}
F_{11}	Avg. best solution	0.00	0.134	0.00	1.58×10^{-04}
	SD	0.00	0.730	0.00	7.94×10^{-05}
	Median Best solution	0.00	0.00	0.00	1.38×10^{-04}
F_{12}	Avg. best solution	0.988	1.72×10^{-02}	0.958	$\mathbf{4.06 \times 10^{-10}}$
	SD	0.210	4.78×10^{-02}	0.167	$\mathbf{9.88 \times 10^{-10}}$
	Median Best solution	1.00	2.43×10^{-32}	1.01	3.57×10^{-11}
F_{13}	Avg. best solution	−**3.00**	−**3.00**	−**3.00**	−**3.00**
	SD	**0.00**	**0.00**	**0.00**	**0.00**
	Median Best solution	−3.00	−3.00	−3.00	−3.00
Avg. rank		**1.2**	2.8	1.9	2.5
Final rank		**1**	4	2	3

8. Non-continuous Rastrigin's Function (multimodal function)

$$F_8(x) = \sum_{i=1}^{n} [y_i^2 - 10\cos(2\pi y_i) + 10]$$

$$\text{where } y_i = \begin{cases} x_i & |x_i| \leqslant 0.5 \\ \frac{round(2x_i)}{2} & |x_i| \geqslant 0.5 \end{cases}$$

9. Ackley's Function (multimodal function)

$$F_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$$

10. Generalized Griewank Function (multimodal function)

$$F_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

11. Weierstrass Function (multimodal function)

$$F_{11}(x) = \sum_{i=1}^{n}\left(\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k(x_i + 0.5))]\right) - n\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k \times 0.5)]$$

where $a = 0.5$, $b = 3$, $k\max = 20$.

12. Generalized Penalized Function (multimodal function)

$$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$$

$$y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leqslant x_i \leqslant a \\ k(-x_i - a)^m & x_i < -a \end{cases}$$

Table 3
Minimization results of rotated and shifted unimodal and multimodal functions (iteration = 5000 and n = 30).

Functions		MPSO	PSO	LMPSO	LPSO
F_{14}	Avg. best solution	**0.00**	190.59	157.75	191.21
	SD	**0.00**	19.06	10.46	9.93
	Median Best solution	**0.00**	190.40	159.66	190.83
F_{15}	Avg. best solution	$\mathbf{9.99 \times 10^{-02}}$	0.390	$\mathbf{9.99 \times 10^{-02}}$	0.573
	SD	1.26×10^{-05}	8.00×10^{-02}	$\mathbf{2.09 \times 10^{-06}}$	7.40×10^{-02}
	Median Best solution	9.99×10^{-02}	0.40	9.99×10^{-02}	0.600
F_{16}	Avg. best solution	28.96	28.80	28.92	**28.76**
	SD	3.2×10^{-02}	0.447	4.8×10^{-02}	3.6×10^{-02}
	Median Best solution	28.97	28.71	28.92	28.74
F_{17}	Avg. best solution	$\mathbf{2.00 \times 10^{-137}}$	$4.91 \times 10^{+04}$	550.59	$4.61 \times 10^{+04}$
	SD	$\mathbf{3.82 \times 10^{-137}}$	$1.14 \times 10^{+04}$	537.93	$9.47 \times 10^{+03}$
	Median Best solution	$1.22 \times 10-^{137}$	$4.84 \times 10^{+04}$	381.96	$4.61 \times 10^{+04}$
F_{18}	Avg. best solution	−370.95	−374.04	**−377.55**	−375.14
	SD	1.47	1.98	1.89	2.08
	Median Best solution	−371.73	−374.34	−377.90	−375.85
F_{19}	Avg. best solution	−119.03	−119.10	−119.05	**−119.11**
	SD	4.60×10^{-02}	6.36×10^{-02}	5.61×10^{-02}	5.13×10^{-02}
	Median Best solution	−119.03	−119.09	−119.04	−119.10
F_{20}	Avg. best solution	135.86	**116.02**	131.52	116.29
	SD	1.43	2.73	1.20	1.71
	Median Best solution	136.10	115.95	131.47	116.44
Avg. rank		2.7	2.4	**2.1**	2.2
Final rank		4	3	**1**	2

Table 4
Comparison of MPSO and LMPSO with PSO and LPSO on Wilcoxon rank sum test ($n = 30$).

Functions	Wilcoxon rank sum test	MPSO		LMPSO	
		PSO	LPSO	PSO	LPSO
F_1	*p*-Value	3.01986e−011	3.01986e−011	2.37682e−007	3.01986e−011
	h-Value	**1**	**1**	**1**	**1**
	zval	−6.6456	−6.6456	−5.16716	−6.6456
F_2	*p*-Value	2.8216e−011	3.01986e−011	0.0268566	3.01986e−011
	h-Value	**1**	**1**	**1**	**1**
	zval	−6.65559	−6.6456	−2.2136	−6.6456
F_3	*p*-Value	3.01986e−011	3.01986e−011	3.01986e−011	3.01986e−011
	h-Value	**1**	**1**	**1**	**1**
	zval	−6.6456	−6.6456	−6.6456	−6.6456
F_4	*p*-Value	3.01986e−011	3.01986e−011	3.01986e−011	3.01986e−011
	h-Value	**1**	**1**	**1**	**1**
	zval	−6.6456	−6.6456	−6.6456	−6.6456
F_5	*p*-Value	0	0	0	0
	h-Value	0	0	0	0
	zval	0	0	0	0
F_6	*p*-Value	3.01986e−011	3.01986e−011	3.01986e−011	3.01986e−011
	h-Value	**1**	**1**	**1**	**1**
	zval	−6.6456	−6.6456	−6.6456	−6.6456
F_7	*p*-Value	1.21178e−012	1.21178e−012	3.16021e−012	3.16021e−012
	h-Value	**1**	**1**	**1**	**1**
	zval	−7.10402	−7.10402	−6.97041	−6.97041
F_8	*p*-Value	1.2098e−012	1.21178e−012	4.35095e−005	6.54865e−004
	h-Value	1	1	−1	−1
	zval	−7.10425	−7.10402	4.08801	3.40781
F_9	*p*-Value	5.15239e−013	1.21178e−012	5.15239e−013	1.21178e−012
	h-Value	**1**	**1**	**1**	**1**
	zval	−7.22122	−7.10402	−7.22122	−7.10402
F_{10}	*p*-Value	5.76763e−009	1.21178e−012	5.76763e−009	1.21178e−012
	h-Value	**1**	**1**	**1**	**1**
	zval	−5.82336	−7.10402	−5.82336	−7.10402
F_{11}	*p*-Value	1.45517e−004	1.21178e−012	1.45517e−004	1.21178e−012
	h-Value	**1**	**1**	**1**	**1**
	zval	−3.7986	−7.10402	−3.7986	−7.10402
F_{12}	*p*-Value	3.01608e−011	3.01986e−011	3.01608e−011	3.01986e−011
	h-Value	−1	−1	−1	−1
	zval	6.64578	6.6456	6.64578	6.6456
F_{13}	*p*-Value	0	0	0	0
	h-Value	0	0	0	0
	zval	0	0	0	0
F_{14}	*p*-Value	1.21178e−012	1.21178e−012	7.77255e−009	5.49405e−011
	h-Value	**1**	**1**	**1**	**1**
	zval	−7.10402	−7.10402	−5.77332	−6.55689
F_{15}	*p*-Value	2.06146e−011	2.82695e−011	2.06146e−011	2.82695e−011
	h-Value	**1**	**1**	**1**	**1**
	zval	−6.7016	−6.65531	−6.7016	−6.65531
F_{16}	*p*-Value	8.48477e−009	3.33839e−011	1.01045e−008	7.38908e−011
	h-Value	−1	−1	−1	−1
	zval	5.75853	6.63081	5.72896	6.51254
F_{17}	*p*-Value	3.01986e−011	3.01986e−011	3.01986e−011	3.01986e−011
	h-Value	**1**	**1**	**1**	**1**
	zval	−6.6456	−6.6456	−6.6456	−6.6456
F_{18}	*p*-Value	1.67797e−008	2.22671e−010	1.01821e−006	2.52369e−005
	h-Value	−1	−1	1	1
	zval	5.6423	6.34483	−4.88809	−4.21267
F_{19}	*p*-Value	1.33668e−005	3.83494e−006	6.20265e−004	1.68132e−004
	h-Value	−1	−1	−1	−1
	zval	4.35401	4.62013	3.42259	3.76264
F_{20}	*p*-Value	3.01986e−011	3.01986e−011	3.01986e−011	3.01986e−011

Table 4 (*continued*)

Functions	Wilcoxon rank sum test	MPSO		LMPSO	
		PSO	LPSO	PSO	LPSO
	h-Value	−1	−1	−1	−1
	zval	6.6456	6.6456	6.6456	6.6456
1 (better)		**13**	**13**	**13**	**13**
0 (same)		2	2	2	2
−1 (worse)		5	5	5	5

Table 5
Minimization results of unimodal and multimodal functions (iteration = 7000 and n = 100).

Functions		MPSO	PSO	LMPSO	LPSO
F_1	Avg. best solution	6.73×10^{-41}	$1.90 \times 10^{+04}$	2.11×10^{-45}	1.73
	SD	3.95×10^{-41}	$1.19 \times 10^{+04}$	2.71×10^{-45}	0.85
	Median Best solution	5.52×10^{-41}	$2.00 \times 10^{+04}$	1.14×10^{-45}	1.56
F_2	Avg. best solution	$\mathbf{7.01 \times 10^{-31}}$	156.33	1.15×10^{-24}	0.130
	SD	$\mathbf{7.77 \times 10^{-31}}$	40.98	8.83×10^{-24}	4.00×10^{-02}
	Median Best solution	3.31×10^{-31}	150.00	9.28×10^{-24}	0.120
F_3	Avg. best solution	$\mathbf{1.46 \times 10^{-164}}$	$1.95 \times 10^{+05}$	3.36×10^{-45}	$2.01 \times 10^{+05}$
	SD	**0.00**	$4.44 \times 10^{+04}$	7.30×10^{-45}	$3.04 \times 10^{+04}$
	Median Best solution	1.48×10^{-164}	$1.99 \times 10^{+05}$	1.05×10^{-45}	$2.03 \times 10^{+05}$
F_4	Avg. best solution	$\mathbf{7.51 \times 10^{-61}}$	55.93	1.78×10^{-24}	67.29
	SD	$\mathbf{9.57 \times 10^{-61}}$	4.25	1.48×10^{-24}	3.87
	Median Best solution	3.02×10^{-61}	56.08	1.50×10^{-24}	67.55
F_5	Avg. best solution	0.00	$2.30 \times 10^{+04}$	0.00	41.07
	SD	0.00	$1.32 \times 10^{+04}$	0.00	10.57
	Median Best solution	0.00	$2.00 \times 10^{+04}$	0.00	39.00
F_6	Avg. best solution	$\mathbf{3.04 \times 10^{-06}}$	115.65	4.61×10^{-04}	1.10
	SD	$\mathbf{2.65 \times 10^{-06}}$	57.99	1.23×10^{-04}	0.226
	Median Best solution	2.46×10^{-06}	109.01	4.53×10^{-04}	1.06
F_7	Avg. best solution	**0.00**	554.29	0.00	548.75
	SD	**0.00**	66.32	0.00	68.80
	Median Best solution	0.00	574.76	0.00	550.67
F_8	Avg. best solution	**0.00**	658.36	0.00	652.04
	SD	**0.00**	84.15	0.00	77.09
	Median Best solution	0.00	657.00	0.00	660.59
F_9	Avg. best solution	$\mathbf{4.44 \times 10^{-15}}$	13.71	$\mathbf{4.44 \times 10^{-15}}$	3.86
	SD	**0.00**	4.48	0.00	0.495
	Median Best solution	4.44×10^{-15}	14.39	4.44×10^{-15}	3.83
F_{10}	Avg. best solution	0.00	159.50	0.00	0.601
	SD	0.00	115.28	0.00	0.162
	Median Best solution	0.00	135.60	0.00	0.599
F_{11}	Avg. best solution	0.00	25.30	0.00	15.72
	SD	0.00	6.60	0.00	6.59
	Median Best solution	0.00	27.25	0.00	14.77
$F12$	Avg. best solution	1.24	$8.53 \times 10^{+06}$	**1.15**	$1.13 \times 10^{+03}$
	SD	3.55×10^{-02}	$4.67 \times 10^{+07}$	$\mathbf{7.48 \times 10^{-02}}$	$1.60 \times 10^{+03}$
	Median Best solution	1.25	3.59	1.16	421.26
F_{13}	Avg. best solution	**−10.00**	−7.77	**−10.00**	−9.00
	SD	**0.00**	1.16	**0.00**	0.579
	Median Best solution	−10.00	−7.60	−10.00	−9.07
Avg. rank		**1.2**	3.3	1.3	2.6
Final rank		**1**	4	2	3

13. Cosine mixture Problem (Multimodal Function)

$$F_{13}(x) = \sum_{i=1}^{n} x_i^2 - 0.1\sum_{i=1}^{n} \cos(5\pi x_i)$$

Table 6
Minimization results of rotated and shifted unimodal and multimodal functions (iteration = 7000 and n = 100).

Functions		MPSO	PSO	LMPSO	LPSO
F_{14}	Avg. best solution	**0.00**	$1.16 \times 10^{+03}$	727.72	$1.06 \times 10^{+03}$
	SD	**0.00**	71.76	49.39	30.03
	Median Best solution	**0.00**	$1.16 \times 10^{+03}$	738.43	$1.05 \times 10^{+03}$
F_{15}	Avg. best solution	$\mathbf{9.99 \times 10^{-02}}$	12.21	$\mathbf{9.99 \times 10^{-02}}$	6.51
	SD	$\mathbf{5.61 \times 10^{-05}}$	5.48	1.54×10^{-04}	0.510
	Median Best solution	9.99×10^{-02}	14.75	9.99×10^{-02}	6.50
F_{16}	Avg. best solution	98.98	$2.02 \times 10^{+09}$	98.92	$2.93 \times 10^{+08}$
	SD	3.28×10^{-02}	$2.30 \times 10^{+09}$	8.19×10^{-02}	$1.32 \times 10^{+08}$
	Median Best solution	98.99	$1.47 \times 10^{+09}$	98.95	$2.65 \times 10^{+08}$
F_{17}	Avg. best solution	$\mathbf{1.62 \times 10^{-179}}$	$7.71 \times 10^{+05}$	1.02×10^{-21}	$7.10 \times 10^{+05}$
	SD	**0.00**	$1.28 \times 10^{+05}$	4.90×10^{-21}	$8.27 \times 10^{+04}$
	Median Best solution	2.96×10^{-179}	$7.53 \times 10^{+05}$	6.00×10^{-21}	$7.21 \times 10^{+05}$
F_{18}	Avg. best solution	−**361.13**	−358.93	−358.37	−358.48
	SD	1.27	0.501	0.589	0.329
	Median Best solution	−361.73	−358.82	−358.35	−358.52
F_{19}	Avg. best solution	−118.67	−118.70	−118.66	−**118.71**
	SD	2.20×10^{-02}	2.55×10^{-02}	1.92×10^{-02}	2.84×10^{-02}
	Median Best solution	−118.67	−118.71	−118.66	−118.71
F_{20}	Avg. best solution	262.64	**211.98**	254.86	217.08
	SD	3.77	7.53	1.86	4.21
	Median Best solution	263.12	212.15	255.13	217.34
Avg. rank		**1.9**	2.9	2.4	2.4
Final rank		**1**	3	2	2

14. Rotated Rastrign Function (multimodal function)

$$F_{14}(x) = \sum_{i=1}^{n} [y_i^2 - 10\cos(2\pi y_i) + 10], \quad y = M \times x$$

15. Rotated Salomon Function (multimodal function)

$$F_{15}(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{n} y_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} y_i^2}, \quad y = M \times x$$

16. Rotated Rosenbrock Function (multimodal function)

$$F_{16}(x) = \sum_{i=1}^{n-1} [(100(y_i^2 - y_{i+1})^2 + (y_i - 1)^2], \quad y = M \times x$$

17. Rotated Elliptic Function (unimodal function)

$$F_{17}(x) = \sum_{i=1}^{n} [(10^6)^{(i-1)/(n-1)} y_i^2, \quad y = M \times x$$

18. Shifted Schwefel's Problem 2.21 (unimodal function)

$$F_{18}(x) = \max_i \{|y_i|, 1 \leqslant i \leqslant n\} + fbias_{18}, \quad y = x - o$$

where $fbias_{18} = -450$.

19. Shifted Rotated Ackley's Function (multimodal function)

$$F_{19}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi z_i\right) + 20 + e + fbias_{19}$$

where $fbias_{19} = -140, z = (x - o) \times M'$ is a linear transformation matrix with condition number = 100.

Table 7
Minimization results of unimodal and multimodal functions (iteration = 9000 and n = 200).

Functions		MPSO	PSO	LMPSO	LPSO
F_1	Avg. best solution	1.43×10^{-45}	$7.93 \times 10^{+04}$	6.31×10^{-58}	712.01
	SD	7.91×10^{-46}	$2.12 \times 10^{+04}$	1.14×10^{-57}	206.13
	Median Best solution	1.26×10^{-45}	$7.50 \times 10^{+04}$	3.58×10^{-58}	728.02
F_2	Avg. best solution	$\mathbf{1.70 \times 10^{-32}}$	555.62	8.36×10^{-30}	39.04
	SD	$\mathbf{1.32 \times 10^{-32}}$	106.35	9.32×10^{-30}	14.23
	Median Best solution	1.39×10^{-32}	544.51	5.95×10^{-30}	35.82
F_3	Avg. best solution	$\mathbf{2.93 \times 10^{-210}}$	$6.43 \times 10^{+05}$	3.32×10^{-56}	$7.67 \times 10^{+05}$
	SD	**0.00**	$1.02 \times 10^{+05}$	8.32×10^{-56}	$8.89 \times 10^{+04}$
	Median Best solution	3.80×10^{-210}	$6.24 \times 10^{+05}$	1.76×10^{-57}	$7.61 \times 10^{+05}$
F_4	Avg. best solution	$\mathbf{3.85 \times 10^{-77}}$	93.21	5.57×10^{-31}	96.04
	SD	$\mathbf{7.01 \times 10^{-77}}$	3.84	6.79×10^{-31}	2.07
	Median Best solution	1.51×10^{-77}	94.01	3.47×10^{-31}	96.34
F_5	Avg. best solution	**0.00**	$7.89 \times 10^{+04}$	0.00	$1.55 \times 10^{+03}$
	SD	**0.00**	$1.85 \times 10^{+04}$	0.00	408.89
	Median Best solution	0.00	$8.01 \times 10^{+04}$	0.00	$1.46 \times 10^{+03}$
F_6	Avg. best solution	$\mathbf{2.61 \times 10^{-6}}$	918.27	3.79×10^{-04}	18.14
	SD	$\mathbf{2.26 \times 10^{-6}}$	281.79	9.69×10^{-05}	7.64
	Median Best solution	2.01×10^{-6}	890.01	3.71×10^{-04}	16.93
F_7	Avg. best solution	0.00	$1.33 \times 10^{+03}$	**0.00**	$1.39 \times 10^{+03}$
	SD	0.00	120.63	**0.00**	151.58
	Median Best solution	0.00	$1.35 \times 10^{+03}$	0.00	$1.39 \times 10^{+03}$
F_8	Avg. best solution	0.00	$1.59 \times 10^{+03}$	0.00	$1.61 \times 10^{+03}$
	SD	0.00	127.48	0.00	119.43
	Median Best solution	0.00	$1.58 \times 10^{+03}$	0.00	$1.62 \times 10^{+03}$
F_9	Avg. best solution	$\mathbf{4.44 \times 10^{-15}}$	18.72	$\mathbf{4.44 \times 10^{-15}}$	10.18
	SD	**0.00**	0.370	**0.00**	1.23
	Median Best solution	4.44×10^{-15}	18.75	4.44×10^{-15}	9.88
F_{10}	Avg. best solution	**0.00**	661.162	**0.00**	8.14
	SD	**0.00**	188.07	**0.00**	2.31
	Median Best solution	0.00	631.17	0.00	7.38
F_{11}	Avg. best solution	**0.00**	89.65	**0.00**	127.13
	SD	**0.00**	9.67	**0.00**	8.94
	Median Best solution	0.00	90.79	0.00	127.98
F_{12}	Avg. best solution	1.20	$4.27 \times 10^{+08}$	1.17	$1.99 \times 10^{+06}$
	SD	3.26×10^{-02}	$2.88 \times 10^{+08}$	3.10×10^{-02}	$1.19 \times 10^{+06}$
	Median Best solution	1.22	$2.56 \times 10^{+08}$	1.17	$1.87 \times 10^{+06}$
F_{13}	Avg. best solution	−**20.00**	−10.28	−**20.00**	−10.97
	SD	**0.00**	1.84	**0.00**	2.00
	Median Best solution	−20.00	−10.18	−20.00	−11.09
Avg. rank		**1.2**	3.1	1.3	2.9
Final rank		**1**	4	2	3

20. Shifted Rotated Weierstrass Function (multimodal function)

$$F_{20}(x) = \sum_{i=1}^{n}\left(\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k(z_i+0.5))]\right) - n\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k \times 0.5)] + fbias_{20}$$

where $a = 0.5$, $b = 3$, $k\max = 20$, $fbias_{20} = 90$, $z = (x - o) \times M'$ is a linear transformation matrix with condition number = 5.

4.2. Comparison of MPSO and LMPSO with PSO and LPSO

MPSO and LMPSO are applied on the minimization functions and the results are compared with PSO and LPSO. Several parameters are set to initial values. Population size is considered as 50 ($N = 50$). C_1 and C_2 in PSO and LPSO are set to 2, and w is linearly decreased from 0.9 to 0.4 [2,8]. Also, the ring topology is considered as the neighborhood structure in lbest model for both LMPSO and LPSO algorithms and the number of neighbors is equal to 2.

The algorithms are independently run 30 times on the functions and the results are averaged and ranked based on the best solution. Also, the standard deviation (SD) and the median of the best solution in the last iteration are reported along

Table 8
Minimization results of rotated and shifted unimodal and multimodal functions (iteration = 9000 and n = 200).

Functions		MPSO	PSO	LMPSO	LPSO
F_{14}	Avg. best solution	**0.00**	$2.76 \times 10^{+03}$	631.12	$2.52 \times 10^{+03}$
	SD	**0.00**	71.76	742.38	53.23
	Median Best solution	0.00	$2.76 \times 10^{+03}$	2.07×10^{-05}	$2.52 \times 10^{+03}$
F_{15}	Avg. best solution	$\mathbf{9.99 \times 10^{-02}}$	32.00	0.107	18.38
	SD	$\mathbf{5.12 \times 10^{-05}}$	3.51	2.54×10^{-02}	1.17
	Median Best solution	9.99×10^{-02}	32.05	9.99×10^{-02}	18.30
F_{16}	Avg. best solution	198.97	$5.82 \times 10^{+10}$	**198.93**	$1.37 \times 10^{+10}$
	SD	4.56×10^{-02}	$2.79 \times 10^{+10}$	0.103	$3.00 \times 10^{+09}$
	Median Best solution	198.99	$5.55 \times 10^{+10}$	198.95	$1.38 \times 10^{+10}$
F_{17}	Avg. best solution	$\mathbf{9.05 \times 10^{-211}}$	$2.57 \times 10^{+06}$	3.70×10^{-47}	$2.31 \times 10^{+06}$
	SD	**0.00**	$2.87 \times 10^{+05}$	2.01×10^{-46}	$2.00 \times 10^{+05}$
	Median Best solution	1.77×10^{-211}	$2.57 \times 10^{+06}$	6.61×10^{-52}	$2.36 \times 10^{+06}$
F_{18}	Avg. best solution	−354.34	−354.75	**−356.04**	−355.83
	SD	0.203	0.575	0.889	0.395
	Median Best solution	−354.24	−354.73	−356.57	−355.93
F_{19}	Avg. best solution	**−118.54**	**−118.54**	−118.53	−118.53
	SD	$\mathbf{1.33 \times 10^{-02}}$	1.44×10^{-02}	1.64×10^{-02}	1.84×10^{-02}
	Median Best solution	−118.54	−118.54	−118.53	−118.53
F_{20}	Avg. best solution	439.57	**435.85**	437.72	437.91
	SD	2.88	**2.58**	3.89	2.76
	Median Best solution	440.41	435.57	439.19	438.37
Avg. rank		2	3	**1.7**	2.7
Final rank		2	4	**1**	3

the dimension (n) 30, 100 and 200 as shown in Tables 2–8. The maximum iteration is 5000 for n = 30, 7000 for n = 100 and 9000 for n = 200 respectively.

Table 2 shows the results of algorithms for unimodal and multimodal functions 1–13 with n = 30. As seen, MPSO and MPSOL have the first and the second rank. Also, MPSO presents the best solution for all functions in the table except in F_{12}. In this function, LPSO has better results. Moreover, MPSO provides the global minimum in the functions F_5, F_7, F_8, F_{10}, F_{11} and F_{13}, and the good near-global optimum in F_1, F_2, F_3, F_4, F_6 and F_9. Furthermore, LMPSO illustrates good performance in unimodal functions and have better convergence rate than PSO and LPSO in the multimodal functions F_7, F_9, F_{10} and F_{11}. Also, the superior convergence rate of MPSO and LMPSO can be depicted from Figs. 6 and 7. As seen in these Figs., MPSO tends to find the global optimum in F_4 and F_{10} faster than other algorithms and obtains the highest accuracy for the functions among the algorithms.

In Table 3, the average results of rotated, shifted and shifted rotated functions 14–20 with n = 30 are demonstrated. Among the algorithms, LMPSO achieves the first rank in this table. As observed, MPSO has significantly better results in

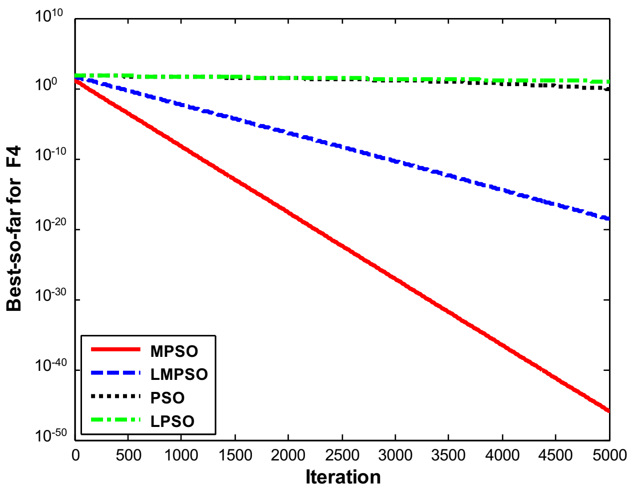

Fig. 6. Convergence performance of MPSO, LMPSO, PSO and LPSO on the unimodal test function F_4 (n = 30).

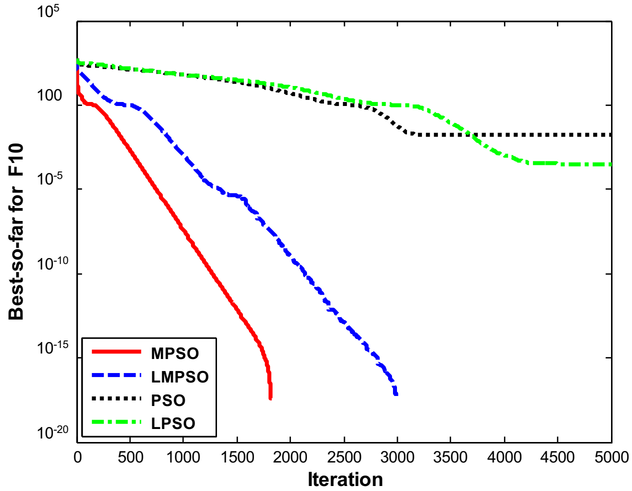

Fig. 7. Convergence performance of MPSO, LMPSO, PSO and LPSO on the multimodal test function F_{10} (n = 30).

F_{14} and F_{17}. For function F_{20}, PSO and LPSO have better performance respectively. Figs. 8 and 9 illustrate the convergence characteristics of the algorithms for obtaining the results on rotated multimodal function F_{14} and shifted unimodal function F_{18}. As shown in these figures, MPSO and LMPSO demonstrate better performance in these functions respectively.

In order to determine whether the results achieved by the algorithms are statistically different from each other, the non-parametric Wilcoxon's rank sum test [32] is conducted between the results obtained by the proposed algorithms, PSO and LPSO. The test is performed only for $n = 30$ as shown in Table 4 because the results of algorithms in some cases are near together in Tables 2 and 3. For $n = 100$ and $n = 200$, the results are quite far from each other and it is easy to judge which algorithm has better performance.

The Wilcoxon's rank sum test returns p-value, h-value and *zval* (the value of the z-statistic computed only for large samples). In Table 4, if p-value is greater than significance level of $\alpha = 0.05$, the h-value is equal to zero; otherwise the value of h-value is considered as "1" or "-1". h-Value = 1 indicates that the case in which proposed algorithm significantly outperforms the compared algorithm with 95% certainty, whereas h-value = 0 implies that the performances of two compared algorithms are not statistically different and h-value = -1 represents the compared algorithm is significantly better than the proposed algorithm. At the end of the table, rows "1 (Better)," "0 (Same)," and "-1 (Worse)"give the number of functions which the MPSO and LMPSO perform significantly better than, almost the same as, and significantly worse than the compared algorithm, respectively. As seen, the results of MPSO and LMPSO in most cases are statistically different and better than PSO and LPSO.

Tables 5 and 6 present the results of the algorithms on the benchmark functions for $n = 100$. Also, Tables 7 and 8 show the performance of algorithms on the functions with $n = 200$. As demonstrated in these tables, PSO and LPSO cannot tune

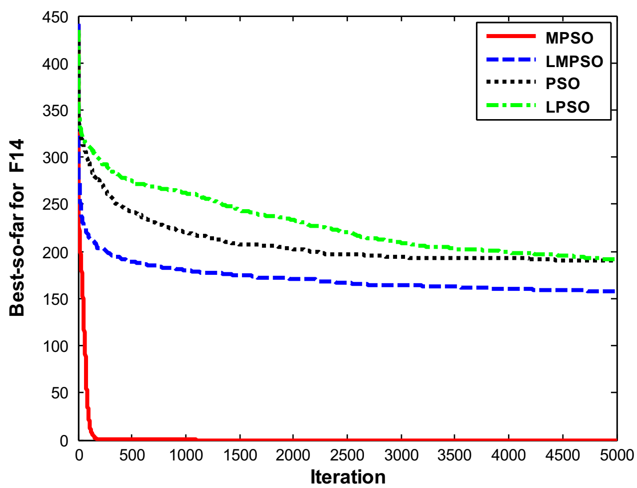

Fig. 8. Convergence performance of MPSO, LMPSO, PSO and LPSO on the rotated multimodal test function F_{14} ($n = 30$).

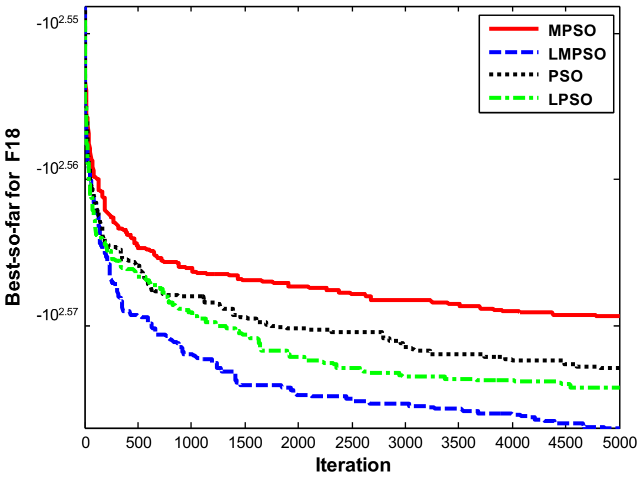

Fig. 9. Convergence performance of MPSO, LMPSO, PSO and LPSO on the shifted unimodal test function F_{18} (n = 30).

themselves and provide the poor results in the majority of functions. In contrast, MPSO and LMPSO achieve the global minimum or the good near-global optimum in most of the functions. Therefore, it could be concluded that MPSO and LMPSO are more powerful and robust than PSO and LPSO to solve unimodal, multimodal functions with high dimension. Also, the less standard deviation of the proposed methods in these tables confirms the superiority of MPSO and LMPSO against PSO and LPSO. Furthermore, it is obvious that the performance of PSO and LPSO decline significantly when the dimension of functions increases.

Figs. 10 and 11 demonstrate the convergence performance of algorithms on F_7 and F_{17} as multimodal and rotated unimodal functions with $n = 100$ respectively. As observed, the proposed algorithms have acquired the global optimum in F_7 and show considerably better results than PSO and LPSO in F_{17}.

Also, the results of algorithms on functions F_{13} and F_{16} with $n = 200$ are illustrated in Figs. 12 and 13. It is observed that MPSO and LMPSO provide global minimum in function F_{13} and have faster convergence rate than PSO and LPSO in F_{16}.

4.3. Comparison of MPSO with other PSO algorithms

In this section, a comparison of MPSO with some well-known PSO algorithms is performed to evaluate the efficiency of the proposed algorithm. The results of MPSO are compared with those of algorithms in [18,23].

At first, 10 unimodal and multimodal high-dimensional benchmark functions of this study, which are common with Ref. [18], are selected for this evaluation. The algorithms of GPSO [2], LPSO [8], FIPS [10] HPSO-TVAC [11], VPSO [9], DMS-PSO [12], CLPSO [13] and APSO [18] are considered as detailed in Table 9. GPSO with global star topology, LPSO with ring

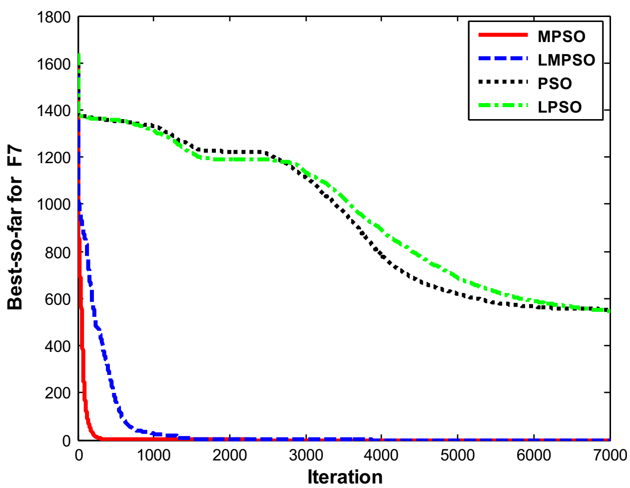

Fig. 10. Convergence performance of MPSO, LMPSO, PSO and LPSO on the multimodal test function F_7 ($n = 100$).

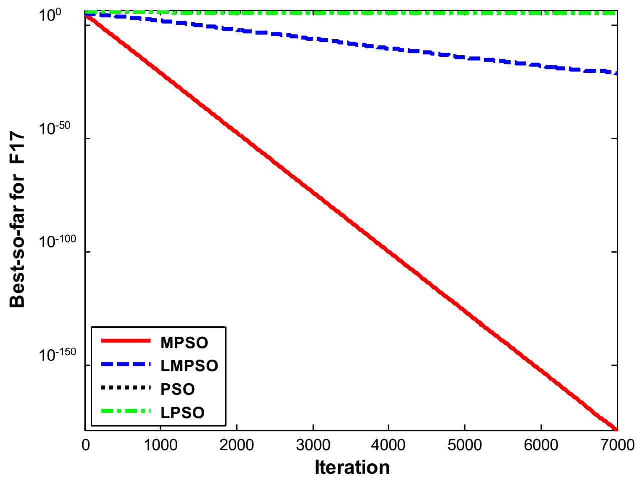

Fig. 11. Convergence performance of MPSO, LMPSO, PSO and LPSO on the rotated unimodal test function F_{17} ($n = 100$).

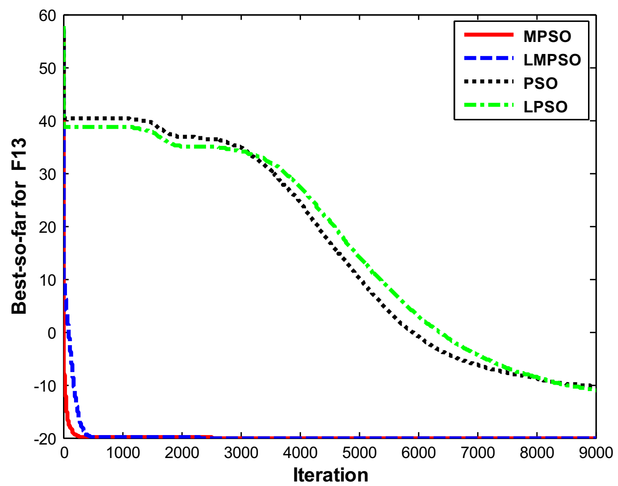

Fig. 12. Convergence performance of MPSO, LMPSO, PSO and LPSO on the multimodal F_{13} ($n = 200$).

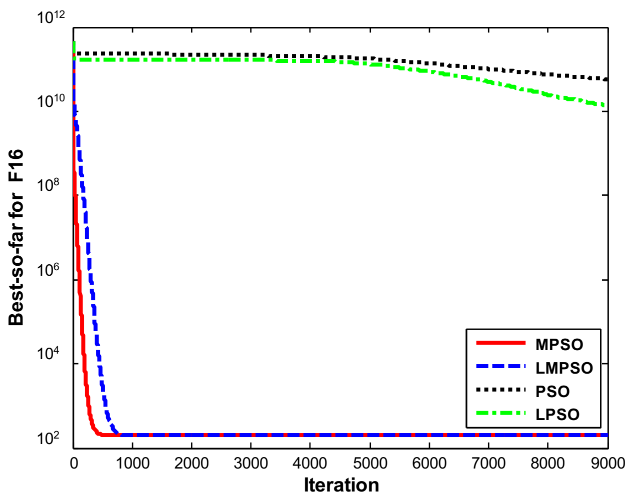

Fig. 13. Convergence performance of MPSO, LMPSO, PSO and LPSO on the rotated multimodal F_{16} ($n = 200$).

Table 9
Some well-known PSOs algorithms in the literature.

Algorithm	Topology	Parameter settings
GPSO	Global Star	ω: 0.9–0.4, $C_1 = C_2 = 2.0$
LPSO	Local Ring	ω: 0.9–0.4, $C_1 = C_2 = 2.0$
FIPS	Local URing	$\chi = 0.729$, $\Sigma ci = 4.1$
HPSO-TVAC	Global Star	ω: 0.9–0.4, C_1: 2.5–0.5, C_2: 0.5–2.5
DMS-PSO	Dynamic Multi-Swarm	ω: 0.9–0.2, $C_1 = C_2 = 2.0$, $m = 3$, $R = 5$
VPSO	Local Von Neumann	ω: 0.9–0.4, $C_1 = C_2 = 2.0$
CLPSO	Comprehensive Learning	ω: 0.9–0.4, $C = 1.49445$, $m = 7$
APSO	Global Star	ω: 0.9, $C_1 = C_2 = 2.0$, δ: random in [0.05,0.1], σ: 1–0.1
QIPSO	Global Star	ω: 0.9–0.4, $C_1 = C_2 = 2.0$
UPSO	Global Star	ω: 0.9–0.4, $C_1 = C_2 = 2.0$, $U = 0.5$
AFPSO	Global Star	ω: 0.9–0.4, C_1, C_2 are based on fuzzy rule [23]
AFPSO-QI	Global Star	ω: 0.9–0.4, C_1, C_2 are based on fuzzy rule [23]

Table 10
Comparison results of eight PSO algorithms [18] with MPSO on 10 benchmark functions (iteration = 200,000, $n = 30$ and $N = 20$).

Function		GPSO	LPSO	FIPS	HPSO-TVAC	DMS-PSO	VPSO	CLPSO	APSO	MPSO
F_1	Best	1.98×10^{-53}	4.77×10^{-29}	3.21×10^{-30}	3.38×10^{-41}	3.85×10^{-54}	5.11×10^{-38}	1.89×10^{-19}	$1.45 \times 10^{-15}0$	**0.00**
	SD	7.08×10^{-53}	1.13×10^{-28}	1.91×10^{-30}	8.50×10^{-41}	1.75×10^{-53}	1.91×10^{-37}	1.49×10^{-19}	$5.73 \times 10^{-15}0$	**0.00**
	Rank	4	8	7	5	3	6	9	2	**1**
F_2	Best	2.51×10^{-34}	2.03×10^{-20}	1.32×10^{-17}	6.9×10^{-23}	2.61×10^{-29}	6.29×10^{-27}	1.01×10^{-13}	5.15×10^{-84}	$\mathbf{1.19 \times 10^{-323}}$
	SD	5.84×10^{-34}	2.89×10^{-20}	7.86×10^{-18}	6.89×10^{-23}	6.6×10^{-29}	8.68×10^{-27}	6.51×10^{-14}	1.44×10^{-83}	**0.00**
	Rank	3	7	8	6	4	5	9	2	**1**
F_3	Best	6.45×10^{-2}	18.60	0.77	2.89×10^{-7}	47.5	1.44	395	1.0×10^{-10}	**0.00**
	SD	9.46×10^{-2}	30.71	0.86	2.97×10^{-7}	56.4	1.55	142	2.13×10^{-10}	**0.00**
	Rank	4	7	5	3	8	6	9	2	**1**
F_5	Best	0.00	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	**0.00**
	Rank	1	1	1	1	1	1	1	1	**1**
F_6	Best	7.77×10^{-3}	1.49×10^{-2}	2.55×10^{-3}	5.54×10^{-2}	1.1×10^{-2}	1.08×10^{-2}	3.92×10^{-3}	4.66×10^{-3}	$\mathbf{1.89 \times 10^{-7}}$
	SD	2.42×10^{-3}	5.66×10^{-3}	6.25×10^{-4}	2.08×10^{-2}	3.94×10^{-3}	3.24×10^{-3}	1.14×10^{-3}	1.7×10^{-3}	$\mathbf{1.87 \times 10^{-7}}$
	Rank	5	8	2	9	7	6	3	4	**1**
F_7	Best	30.7	34.90	29.98	2.39	28.1	34.09	2.57×10^{-11}	5.8×10^{-15}	**0.00**
	SD	8.68	7.25	10.92	3.71	6.42	8.07	6.64×10^{-11}	1.01×10^{-14}	**0.00**
	Rank	7	9	6	4	5	8	3	2	**1**
F_8	Best	15.5	30.40	21.33	35.91	1.83	32.8	0.167	4.14×10^{-16}	**0.00**
	SD	7.4	9.23	9.46	9.49	2.65	6.49	0.379	1.45×10^{-15}	**0.00**
	Rank	5	7	6	9	4	8	3	2	**1**
F_9	Best	1.15×10^{-14}	1.85×10^{-14}	7.69×10^{-15}	2.06×10^{-10}	8.52×10^{-15}	1.14×10^{-14}	2.01×10^{-12}	1.11×10^{-14}	$\mathbf{4.44 \times 10^{-15}}$
	SD	2.27×10^{-15}	4.80×10^{-15}	9.33×10^{-16}	9.45×10^{-10}	1.79×10^{-15}	3.48×10^{-15}	9.22×10^{-13}	3.55×10^{-15}	**0.00**
	Rank	6	7	2	9	3	5	8	4	**1**
F_{10}	Best	2.37×10^{-2}	1.10×10^{-2}	9.04×10^{-4}	1.07×10^{-2}	1.31×10^{-2}	1.31×10^{-2}	6.45×10^{-13}	1.67×10^{-2}	**0.00**
	SD	2.57×10^{-15}	1.60×10^{-2}	2.78×10^{-3}	1.14×10^{-2}	1.73×10^{-2}	1.35×10^{-2}	2.07×10^{-12}	2.41×10^{-2}	**0.00**
	Rank	9	5	3	4	7	6	2	8	**1**
F_{12}	Best	1.04×10^{-2}	2.18×10^{-30}	1.22×10^{-31}	7.07×10^{-30}	$\mathbf{2.05 \times 10^{-32}}$	3.46×10^{-3}	1.59×10^{-21}	3.76×10^{-31}	1.07
	SD	3.16×10^{-2}	5.14×10^{-30}	4.85×10^{-32}	4.05×10^{-30}	$\mathbf{8.12 \times 10^{-33}}$	1.89×10^{-2}	1.93×10^{-21}	1.2×10^{-30}	0.225
	Rank	8	4	2	5	**1**	7	6	3	9
Avg. rank		5.2	6.3	4.2	5.5	4.3	5.8	5.3	3.0	**1.8**
Final rank		5	9	3	7	4	8	6	2	**1**

neighborhood, and VPSO with Von Neumann neighborhood are regarded as standard PSOs. The topology used in HPSO-TVAC and APSO is global star. DMS-PSO and CLPSO employ Dynamic Multi-Swarm and Comprehensive Learning respectively. The computational results of the algorithms are directly from [18] as shown in Table 10. In this table, the maximum iteration has been considered as 2×10^5 also; n and N are equal to 30 and 20 respectively. The parameter configurations of algorithms have been set according to their corresponding references. MPSO is independently run 30 times and the average best solution and the SD are presented in the table. As seen, MPSO has the first rank among the algorithms and obtains the global minimum in the functions of F_1, F_3, F_5, F_7, F_8 and F_{10}, and gives the good near-global optimum solution in functions F_2, F_6 and F_9. Also, MPSO has the best SD among the algorithms except in F_{12}. In the function, DMS-PSO represents a better solution.

Table 11
Comparison results of seven PSO algorithms [23] with MPSO on six benchmark functions (iteration = 10,000, $n = 30$ and $N = 30$).

Function		GPSO	QIPSO	UPSO	FIPS	CLPSO	AFSO	AFSO-Q1	MPSO
F_7	Best	52.30	25.61	59.40	106.1	74.39	17.93	15.69	**0.00**
	SD	27.35	15.98	58.05	30.54	9.77	5.63	4.47	**0.00**
	Rank	5	4	6	8	7	3	2	**1**
F_{11}	Best	0.534	36.38	8.70	6.40	1.39×10^{-03}	4.52×10^{-03}	1.50×10^{-03}	**0.00**
	SD	1.74	4.66	3.08	3.04	3.28×10^{-04}	9.20×10^{-03}	3.48×10^{-03}	**0.00**
	Rank	5	8	7	6	2	4	3	**1**
F_{14}	Best	320.2	317.5	309.5	434.1	263.3	266.3	253.3	**0.00**
	SD	14.70	23.24	25.88	34.99	11.96	12.00	12.63	**0.00**
	Rank	7	6	5	8	3	4	2	**1**
F_{15}	Best	17.03	15.20	14.29	26.60	11.94	10.38	8.46	$\mathbf{9.87 \times 10^{-02}}$
	SD	2.55	1.32	2.15	1.42	1.37	1.38	0.948	$\mathbf{6.79 \times 10^{-03}}$
	Rank	7	6	5	8	4	3	2	**1**
F_{19}	Best	−119.10	−119.10	−119.10	**−119.90**	−119.00	−119.70	−119.80	−119.10
	SD	7.09×10^{-02}	5.68×10^{-02}	5.82×10^{-02}	$\mathbf{3.24 \times 10^{-02}}$	3.78×10^{-02}	4.28×10^{-02}	3.85×10^{-02}	5.45×10^{-02}
	Rank	4	4	4	**1**	5	3	2	4
F_{20}	Best	115.90	121.90	**113.20**	113.60	118.30	123.20	123.10	136.00
	SD	2.90	4.90	6.14	3.63	2.40	2.25	3.01	1.63
	Rank	3	5	1	2	4	7	6	8
Avg. rank		5.2	5.5	4.7	5.5	4.2	4	2.8	**2.7**
Final rank		6	7	5	7	4	3	2	**1**

Then in the next step, six multimodal functions, which are common with Ref. [23], are chosen and the results of MPSO is evaluated in comparison with seven algorithms of GPSO [2], QIPSO [16], UPSO [25], FIPS, CLPSO, AFSO [23] and AFSO-Q1 [23] as detailed in Table 9. Table 11 illustrates their performance where, maximum iteration is 10,000 also; n and N are set to 30 and 30 respectively. As seen, MPSO shows better performance and has the first rank. In other words, MPSO gives the global minimum in F_7, F_{11} and F_{14} and has good results in F_{15}. In F_{19} and F_{20}, FIPS and UPSO have better results respectively.

Therefore, it is worth saying that the proposed method has considerably better performance than the other well-known PSO algorithms in unimodal and multimodal high-dimensional functions.

5. Conclusion

In this paper, an improved scheme of PSO called Median-oriented Particle Swarm Optimization (MPSO) has been introduced to enhance the performance of PSO. The algorithm is based on standard PSO and the median position, the worst and the median fitness values of particles. MPSO is a global search algorithm with several advantages which make the algorithm convenient to use; easy to implement, insensitive to variables size, easily parallelized for concurrent processing, and no need for any algorithm-specific parameters are among the benefits of MPSO. Therefore, it leads to the conclusion that MPSO requires only common controlling parameters such as the number of generation and population size.

The proposed algorithm has been implemented for global (MPSO) and local (LMPSO) topology. To evaluate MPSO and LMPSO, a set of standard benchmarks including unimodal and multimodal functions have been employed along with the function dimensions ranging 30–200. The average results of algorithms on the functions have been compared with PSO (global topology) and LPSO (local topology) algorithms. Experimental results show that MPSO and LMPSO have better performance than PSO and LPSO in terms of the quality of the final solutions and the convergence rate especially in the dimensions of 100 and 200.

Moreover, MPSO has been compared with several well-known PSO algorithms in the literature. The results indicate that the proposed algorithm presents better results than the other PSO algorithms and has the first rank among them.

Acknowledgments

This work is supported by The Ministry of Higher Education (MOHE) under Long Term Research Grant Scheme (LRGS/TD/2011/UTM/ICT/03 – Vote 4L805). The authors would like to thank Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM) for the support in R & D, Soft Computing Research Group (SCRG) for the inspiration in making this study a success. The authors would also like to thank the anonymous reviewers who have contributed enormously to this work.

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
[2] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of IEEE International Conference on Evolutionary Computation, 1998, pp. 69–73.

[3] T.H. Kim, I. Maruta, T. Sugie, Robust PID controller tuning based on the constrained particle swarm optimization, Automatica 44 (2008) 1104–1110.
[4] Q. Kang, L. Wang, Q.D. Wu, A novel ecological particle swarm optimization algorithm and its population dynamics analysis, Appl. Math. Comput. 205 (2008) 61–72.
[5] Y. Jiang, T. Hu, C.C. Huang, X. Wu, An improved particle swarm optimization algorithm, Appl. Math. Comput. 193 (1) (2007) 231–239.
[6] X. Yang, J. Yuan, J. Yuan, H. Mao, A modified particle swarm optimizer with dynamic adaptation, Appl. Math. Comput. 189 (2) (2007) 1205–1213.
[7] Z. Cui, X. Cai, J. Zeng, G. Sun, Particle swarm optimization with FUSS and RWS for high dimensional functions, Appl. Math. Comput. 205 (2008) 98–108.
[8] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of IEEE International Conference on Evolutionary Computation, vol. 2, 2002, pp. 1671–1676.
[9] J. Kennedy, R. Mendes, Neighborhood topologies in fully informed and best-of-neighborhood particle swarms, IEEE Trans. Syst. Man Cybernet. Part C 36 (4) (2006) 515–519.
[10] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler maybe better, IEEE Trans. Evol. Comput. 8 (3) (2004) 204–210.
[11] A. Ratnaweera, S. Halgamuge, H. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.
[12] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: Proceedings of IEEE Swarm Intelligence Symposium, 2005, pp. 124–129.
[13] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[14] P.S. Andrews, An investigation into mutation operators for particle swarm optimization, in: Proceedings of IEEE International Congress on Evolutionary Computation, 2006, pp. 1044–1051.
[15] Y.P. Chen, W.C. Peng, M.C. Jian, Particle swarm optimization with recombination and dynamic linkage discovery, IEEE Trans. Syst. Man Cybernet. Part B 37 (6) (2007) 1460–1470.
[16] M. Pant, T. Radha, V.P. Singh, A new particle swarm optimization with quadratic interpolation, in: Proceedings of IEEE International Conference on Computational Intelligence and Multimedia Applications, 2007, pp. 55–60.
[17] P.J. Angeline, Using selection to improve particle swarm optimization, in: Proceedings of IEEE International Conference on Evolutionary Computation, 1998, pp. 84–89.
[18] Z.-H. Zhan, J. Zhang, Y. Li, H.H. Chung, Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybernet. Part B 39 (6) (2009) 1362–1381.
[19] Y.H. Shinn, S.L. Hung, H.L. Weei, J.H. Shinn, Orthogonal particle swarm optimization and its application to task assignment problems, IEEE Trans. Syst. Man Cybernet. Part A 38 (2) (2008) 288–298.
[20] Z.-H. Zhan, J. Zhang, Y. Li, Y.-H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (6) (2011) 832–847.
[21] B. Alatas, E. Akin, O. Bedri, Chaos embedded particle optimization algorithms, Chaos Solitons Fractals 40 (4) (2008) 1715–1734.
[22] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[23] Y.-T. Juang, S.-L. Tung, H.-C. Chiu, Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions, Inf. Sci. 181 (2011) 4539–4549.
[24] Q. Luo, D. Yi, A co-evolving framework for robust particle swarm optimization, Appl. Math. Comput. 199 (2) (2008) 611–622.
[25] K.E. Parsopoulos, M.N. Vrahatis, UPSO: a unified particle swarm scheme, in: Lecture Series on Computer and Computational Sciences, vol. 1, 2004, pp. 868–873.
[26] W.F. Gao, S.Y. Liu, L.L. Huang, Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique, Commun. Nonlinear Sci. Numer. Simul. 17 (11) (2012) 4316–4327.
[27] L. Brand, Differential and Difference Equations, John Wiley and Sons, New York, 1966.
[28] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.
[29] K. Deep, M. Thakur, A new crossover operator for real coded genetic algorithms, Appl. Math. Comput. 188 (2007) 895–911.
[30] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, 2005, pp. 1–50.
[31] R. Salomon, Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions, BioSystems 39 (3) (1996) 263–278.
[32] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bull. 1 (6) (1945) 80–83.