Expert Systems With Applications 152 (2020) 113353

Contents lists available at ScienceDirect

Expert Systems With Applications

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

# A modified particle swarm optimization using adaptive strategy

Hao Liu*, Xu-Wei Zhang, Liang-Ping Tu

*School of Science, University of Science and Technology Liaoning, Anshan 114051, PR China*

A R T I C L E I N F O

*Article history:*
Received 12 August 2019
Revised 31 December 2019
Accepted 3 March 2020
Available online 4 March 2020

*Keywords:*
Particle swarm optimization
Chaos
Stochastic learning
Mainstream learning
Adaptive strategy

A B S T R A C T

In expert systems, complex optimization problems are usually nonlinear, nonconvex, multimodal and discontinuous. As an efficient and simple optimization algorithm, particle swarm optimization(PSO) has been widely applied to solve various real optimization problems in expert systems. However, avoiding premature convergence and balancing the global exploration and local exploitation capabilities of the PSO remains an open issue. To overcome these drawbacks and strengthen the ability of PSO in solving complex optimization problems, a modified PSO using adaptive strategy called MPSO is proposed. In MPSO, in order to well balance the global exploration and local exploitation capabilities of the PSO, a chaos-based non-linear inertia weight is proposed. Meanwhile, to avoid the premature convergence, stochastic and mainstream learning strategies are adopted. Finally, an adaptive position updating strategy and terminal replacement mechanism are employed to enhance PSO's ability to solve complex optimization problems in expert systems. 30 complex CEC2017 benchmark functions are utilized to verify the promising performance of MPSO, experimental results and statistical analysis indicate that MPSO has competitive performance compared with 16 state-of-the-art algorithms. The source code of MPSO is provided at https://github.com/lhustl/MPSO .

© 2020 Elsevier Ltd. All rights reserved.

## 1. Introduction

In expert system, many practical problems can be modeled as optimization tasks, such as power system (Dosoglu, Guvenc, Duman, Sonmez, & Kahraman, 2018; Lai, Illindala, & Subramaniam, 2019; Thurner et al., 2018), path planning (FarzanehKaloorazi, Bonev, & Birglen, 2018; Raineri, Ronchini, Perri, & Bianco, 2020) and water supply system design (Han, Lu, & Qiao, 2017), operation management (Hamdi, Ghorbel, Masmoudi, & Dupont, 2018) and satellite network (Huang, Liu, Su, & Wang, 2018). Because of the wide applications of expert system, complex optimization problems are generally nonlinear, uncertain, multimodal, discontinuous and high-dimensional. Therefore, high performance optimization algorithm has become a research hotspot.

Particle swarm optimization (PSO) is a population-based intelligent optimization algorithm proposed by (Kennedy & Eberhart, 1995), which has attracted more and more attention from researchers and has been successfully applied to solve many real optimization problems in expert systems, which has attracted more and more attention from researchers, such as the vehicle routing problem (Marinakis, Marinaki, & Migdalas, 2019), data classification (Alswaitti, Albughdadi, & Isa, 2018; Carneiro, Cheng, Zhao, & Jin, 2019), knapsack problem (Lin, Guan, Li, & Feng, 2019b), flexible job-shop scheduling (Nouiri, Bekrar, Jemai, Niar, & Ammari, 2018), feature selection (Wu, Ma, Fan, Xu, & Shen, 2019a), deep neural network (Geng, 2018; Xue, Tang, & Liu, 2019) and many other real-world problems in expert systems (Mahmoodian, Jabbarzadeh, Rezazadeh, & Barzinpour, 2019; Wu, Wu, & Liu, 2019b).

Although PSO is of simple principle, easy implementation and higher convergence rate, it still has some shortcomings, such as premature convergence and poor global search ability. Researchers have proposed various modified versions of PSO to overcome the deficiencies. Generally speaking, there are five strategies to fulfill these targets:

(1) Tuning control parameters. As for the inertial weight ($w$), time-varying inertia weights (Beheshti, Shamsuddin, & Hasan, 2015; Yang, Gao, Liu, & Song, 2015), adaptive inertial weight (Agrawal & Tripathi, 2019; Nagra, Han, & Ling, 2019b; Shi & Eberhart, 2001), rand inertial weight (Eberhart & Shi, 2001) and chaotic dynamic weight (Chen, Zhou, & Liu, 2018a; Xu, Rong, Trovati, Liptrott, & Bessis, 2018) can enhance the performance of PSO. Concerning the cognitive acceleration coefficient ($c_1$) and the social acceleration coefficient ($c_2$), the time-varying acceleration coefficients (Ratnaweera. & Halgamuge., 2004) are widely used.

* Corresponding author.
*E-mail addresses:* liuhustl@sina.cn (H. Liu), tuliangping@ustl.edu.cn (L.-P. Tu).

https://doi.org/10.1016/j.eswa.2020.113353
0957-4174/© 2020 Elsevier Ltd. All rights reserved.

(2) Hybrid PSO, which hybridizes other heuristic operators to increase population diversity. The genetic operators have been hybridized with PSO, such as selection operator (Angeline, 1998), crossover operator (Chen et al., 2018b; Engelbrecht, 2016) and mutation operator (Chen et al., 2017; Dong, Kang, & Zhang, 2017). Similarly, ant colony optimization (Jindal & Bedi, 2018), differential evolution algorithm (Wang, Li, & Yang, 2019), artificial immune system (Krohling & Mendel, 2013), tabu search (Nakano, Ishigame, & Yasuda, 2010), local search strategy (Liang & Suganthan, 2005), artificial bee colony (El-Abd, 2013), firefly algorithm (Aydilek, 2018) and gravitational search algorithm (Nagra, Han, Ling, & Mehta, 2019a) have been introduced into PSO to make full use of the advantages of each algorithm.

(3) Changing the topological structure. The neighbourhood is the set of particles with which a given particle shares information. The global and local versions of PSO are the main type of swarm topologies. The global version converges faster with the disadvantage of trapping into a local optima, while the local version can obtain a better solution with slower convergence (Kennedy, 1999). Many different neighborhood topologies have been devised for PSO including Star, Ring, Wheel, Pyramid and Von Neumann structures (Engelbrecht, 2005). A surrogate-assisted hierarchical particle swarm optimizer was proposed for high-dimensional problems (Yu, Tan, Zeng, Sun, & Jin, 2018). A local optima topology structure based on the comprehensive learning PSO was proposed to enlarge the particle's search space and increase the convergence speed with a certain probability (Zhang, Huang, & Zhang, 2019a). A ring topology in exemplar generation was adopted to enhance diversity and exploration ability of the genetic learning particle swarm optimization (Lin, Sun, Yu, Wu, & Tang, 2019a). Zhang et al. (2019b) proposed a terminal crosser and steering based PSO (TCSPSO), which effectively enhanced PSO's ability to solve complex optimization problems and was used to solve several engineering design problems in expert systems.

(4) Eliminating the velocity formula. After researching the convergence property of PSO, the Bare-bones PSO (BPSO) was proposed (Kennedy, 2003). It must be emphasized that BPSO does not tune control parameters by discarding the velocity item of PSO, i.e., it is parameter-free. Some variants of BPSO have been proposed to enhance the performance of BPSO (Liu, Ding, & Wang, 2014; Liu, Xu, Ding, & Li, 2015; Omran, Engelbrecht, & Salman, 2009; Vafashoar & Meybodi, 2019).

(5) Changing the learning Strategy. The comprehensive learning PSO (CLPSO) (Liang, Qin, Suganthan, & Baskar, 2006) was proposed and was further developed heterogeneous CLPSO to enhance the exploitation and exploration of CLPSO (Lynn & Suganthan, 2015). A social learning PSO (Zhang, Wang, Kang, & Cheng, 2019c) was proposed, in which the social learning mechanism was introduced into PSO to improve the global search ability and convergence performance. A dimensional learning PSO algorithm (Xu et al., 2019) was proposed, in which a dimensional learning strategy is adopted to construct a learning exemplar for each particle. An adaptive learning PSO algorithm (Wang et al., 2018) was devised, in which a self-learning based candidate generation strategy was introduced to ensure the exploration ability, and a competitive learning based prediction strategy was employed to guarantee exploitation ability. Wang, Liu, Yu, and Tu (2020) successfully combined a variant of PSO with cellular automata and used it to optimize neural network in expert systems.

Although these classic PSO variants have been mature and successfully applied to solve real optimization problems in expert systems, but there are still problems of premature convergence and poor performance on complex optimization problems in expert systems. In addition, with the development of human society, the real optimization problems in expert system are more and more complex, and tend to be high-dimensional, multi-constraint, multimodal, and other trends, PSO and other classic optimization algorithms are facing new severe challenges. In the face of this situation, how to improve PSO to improve its ability to solve increasingly complex optimization problems in expert systems has become a research hotspot.

In this study, to strength the performance of PSO in solving complex optimization problems with expert systems and balance exploration and exploitation better, a modified PSO, called MPSO is proposed. In MPSO, there are four contributions, which are described below.

(1) A chaos-based non-linear inertia weight is proposed to balance exploration and exploitation better by enlarging or shrinking the search step.
(2) Stochastic and mainstream learning strategies are devised to replace the self and global learning strategies, which can effectively enhance the diversity of population and avoid premature convergence. And then improve the ability of the algorithm to solve complex optimization problems in expert systems.
(3) An adaptive position updating strategy is introduced to further balance the exploration and exploitation process.
(4) Inspired by the survival of the fittest rule in nature, a terminal replacement mechanism is adopted to enhance the convergence precise of MPSO, and the ability of the algorithm to solve the real optimization problems of expert systems is further improved.

In additions, to verify the feasibility and validity of the proposed MPSO algorithm, 30 well-known benchmark functions from CEC2017 are employed to test MPSO convergence performance. MPSO is compared with 16 excellent PSO variants and other evolutionary algorithms developed recently. Experimental results and corresponding statistical analysis indicate that MPSO outperforms other 16 algorithms for majority benchmark functions. In summary, the proposed MPSO is successful, it has a competitive convergence performance

The remainder of this paper is presented as follows: the canonical PSO is presented in Section 2. Section 3 describes the proposed MPSO algorithm, which shows better convergence performance. The simulation results and comparisons of the approaches are shown in Section 4. Finally, conclusion and the future work are presented in Section 5.

## 2. Particle swarm optimization (PSO)

As a kind of swarm intelligent optimization algorithms, PSO was proposed by Eberhart and Kennedy in 1995. It uses a simple mechanism that mimics swarm behaviour in birds foraging to guide the particles to search for global optimum. So that PSO relies on the motion of the particles in search space to find the optimal value. In this paper, the dimension of search space is denoted as $D$, and the position and velocity of the $i$-th particle at time $t$ are recorded as $\{X_i^1(t), X_i^2(t), \ldots, X_i^D(t)\}$, and $\{V_i^1(t), V_i^2(t), \ldots, V_i^D(t)\}$, $i = 1, 2, \ldots, N$, respectively. $N$ is the number of particles in the population. The velocity and position update formulas of particle $i$ are represented as follows:

$$V_i(t+1) = \omega(t)V_i(t) + c_1 \cdot r_1(Pbest_i(t) - X_i(t)) + c_2 \cdot r_2(Gbest(t) - X_i(t)) \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

Where $r_1$ and $r_2$ are two random variables which are uniformly distributed in (0,1). $c_1 = c_2 = 2$, referred to as the acceleration factors. Personal best and global best are defined as Eqs. (3) and (4), respectively.

$$Pbest_i(t) = argmin\{fit(X_i(1)), fit(X_i(2)), \ldots, fit(X_i(t))\} \tag{3}$$

$$Gbest(t) = argmin\{fit(Pbest_1(t)), fit(Pbest_2(t)), \ldots, fit(Pbest_N(t))\} \tag{4}$$

In Eq. (1), $\omega$, called inertia weight, is a linear decreasing variable. $\omega$ is generated by Eq. (5).

$$\omega(t) = \omega_{max} - \frac{(\omega_{max} - \omega_{min})}{T_{Max}} t \tag{5}$$

Where $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$. $T_{Max}$ is the maximum number of iterations, $t$ is the number of current iteration.

Obviously, the particle positions in the population are varied with the increase of the number of iterations $t$ through Eqs. (1) and (2). In addition, $Pbest_i(t) - X_i(t)$ is called self-cognition. $Gbest(t) - X_i(t)$ is called social cognition.

## 3. A modified particle swarm optimization using adaptive strategy

In order to avoid the premature convergence phenomenon of PSO and improve its performance on complex problems, a modified particle swarm optimization using adaptive strategy, called MPSO, is proposed in this paper. In MPSO, the four major highlights are described below.

### 3.1. Chaotic-based inertia weight

As one of the parameters in PSO, inertia weight ($\omega$) can provide particles with dynamic adjustment ability in different environments, thus realize the balance between exploration and exploitation. Therefore, it plays an important role in PSO. Generally, linear method of inertia weight is adopted, however, the non-linear inertia weight has stronger fitting and simulation ability.

As a non-linear mapping, chaos generates random numbers with good randomness and disorder, so it has been widely used in the field of evolutionary computation (Tharwat, Elhoseny, Hassanien, Gabel, & Kumar, 2019). Logistic is a well-known chaotic mapping, which can generate random numbers between 0 and 1. It is defined by Eq. (6). In this paper, the logistic chaos is introduced into the inertia weight, thus a non-linear inertia weight is constructed, which is defined by Eq. (7), and the corresponding simulation diagram is shown in Fig. 1.

$$r(t+1) = 4r(t)(1 - r(t)), \; r(0) = rand \tag{6}$$

where $r_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$

$$\omega(t) = r(t) \cdot \omega_{min} + \frac{(\omega_{max} - \omega_{min}) \cdot t}{T_{max}} \tag{7}$$

where $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$. $r(t)$ is a random number generated by the logistic chaotic.

From Fig. 1, it can be seen that the inertia weight $\omega$ is a nonlinear form of value taking and is fluctuating in the whole iteration process, and has strong volatility. This kind of nonlinear inertia weight has stronger fitting and simulation capabilities and can balance well the global exploration and local exploitation abilities of particles. In addition, $\omega$ with wave characteristics can make the motion of particles more random, so that particles are not easy to fall into local optimum.

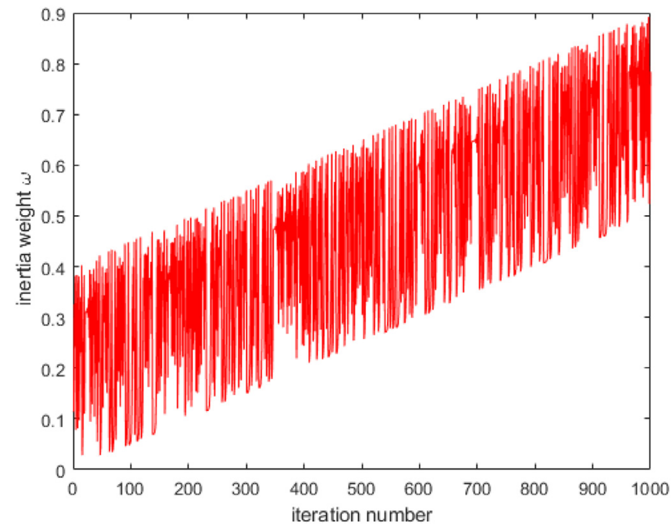



**Fig. 1.** simulation diagram of inertia weight.

### 3.2. Stochastic and mainstream learning strategies

As we know that personal and global learning strategies are adopted in PSO, particles learn from their personal best (*Pbest*) and the global best (*Gbest*) to update their velocities and positions. This learning strategy makes PSO have the advantages of fast convergence, high reliability and strong global exploration ability. However, this strategy leads to the shortcomings of PSO such as premature convergence and poor performance on complex problems. To avoid these issues, stochastic and mainstream learning strategy are introduced to replace the personal and global learning strategies.

Stochastic learning, which gives particles the ability to learn from other excellent individuals in the population, makes the movement of particles more diverse. Therefore, the shortcoming of PSO algorithm that is easy to premature convergence is overcome by enhancing the diversity of population. The strategy is described by Eqs. (8) and (9). At each iteration, two mutually different personal best particles are randomly selected from the population, and the better personal best will be considered as a candidate personal best solution (*CPbest*), then, the better solution, compared the current personal best ($Pbest_i$) with the *CPbest* by their fitness value, will be the final stochastic personal best (*SPbest*). In this way, particles will learn from *SPbest* to update their velocities.

$$CPbest(t) = argmin\{fit(Pbest_a(t)), \; fit(Pbest_b(t))\}, \quad a \neq b \in \{1, \; 2, \; \cdots, \; N\} \tag{8}$$

$$SPbest_i(t) = \begin{cases} CPbest(t) & fit(CPbest) < fit(Pbest_i) \\ Pbest_i(t) & otherwise \end{cases} \tag{9}$$

In PSO and major variants of PSO, each particle updates their position by learning from the personal best experience (*Pbest*) and the population best experience (*Gbest*), so that these algorithms suffer some potential problems, such as diversity quickly lost and premature convergence. Therefore, designing an effective learning strategy to avoid these drawbacks and to improve the convergence performance is an urgent issue for PSO research. As we all known, the mass of society is influenced not only by the best individuals but also by some mainstream ideas. To simulate the social phenomenon, a global particle, denoted as *Mbest*, is defined as the mean of the personal best positions of all particles. That is

$$Mbest(t) = mean\{Pbest_1, Pbest_2, \ldots, Pbest_N\}$$

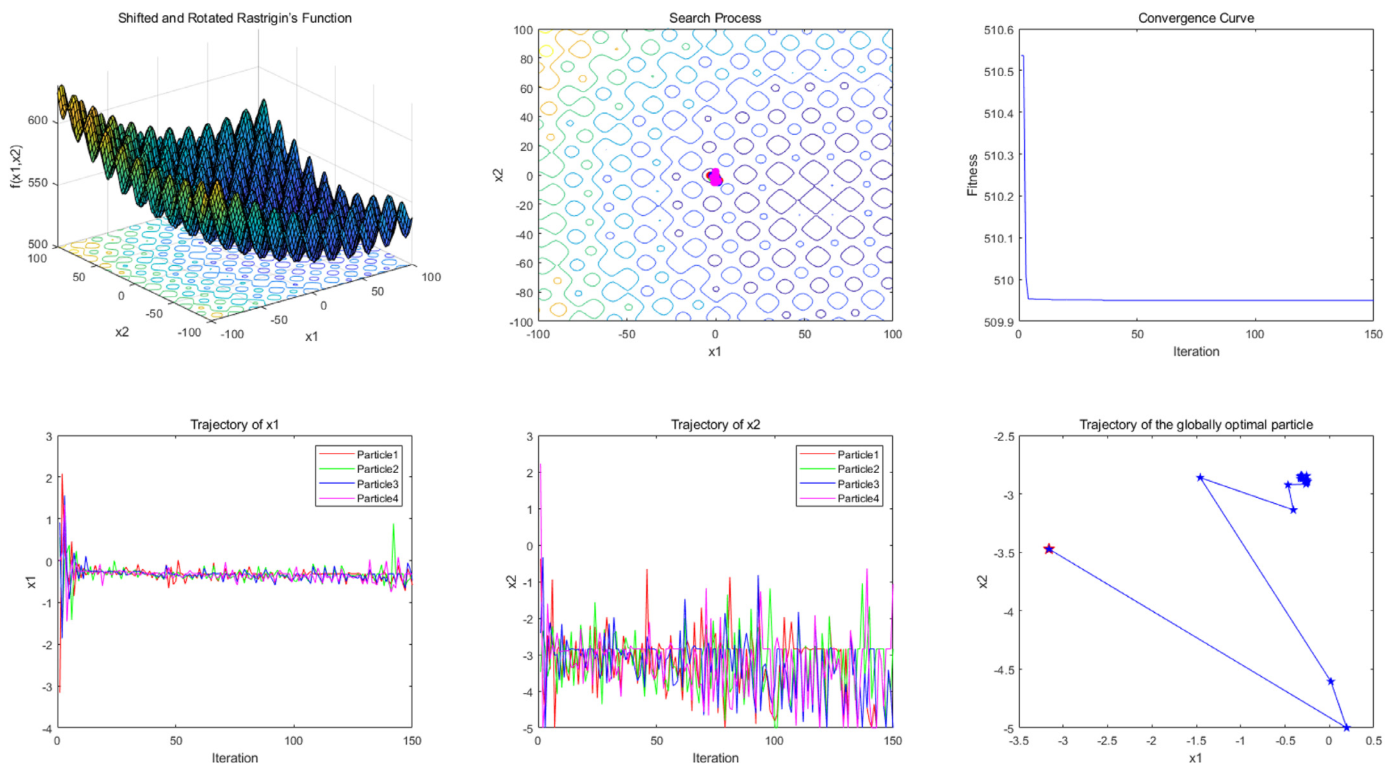



**Fig. 2.** Shifted and Rotated Rastrigin's function $f_5$.

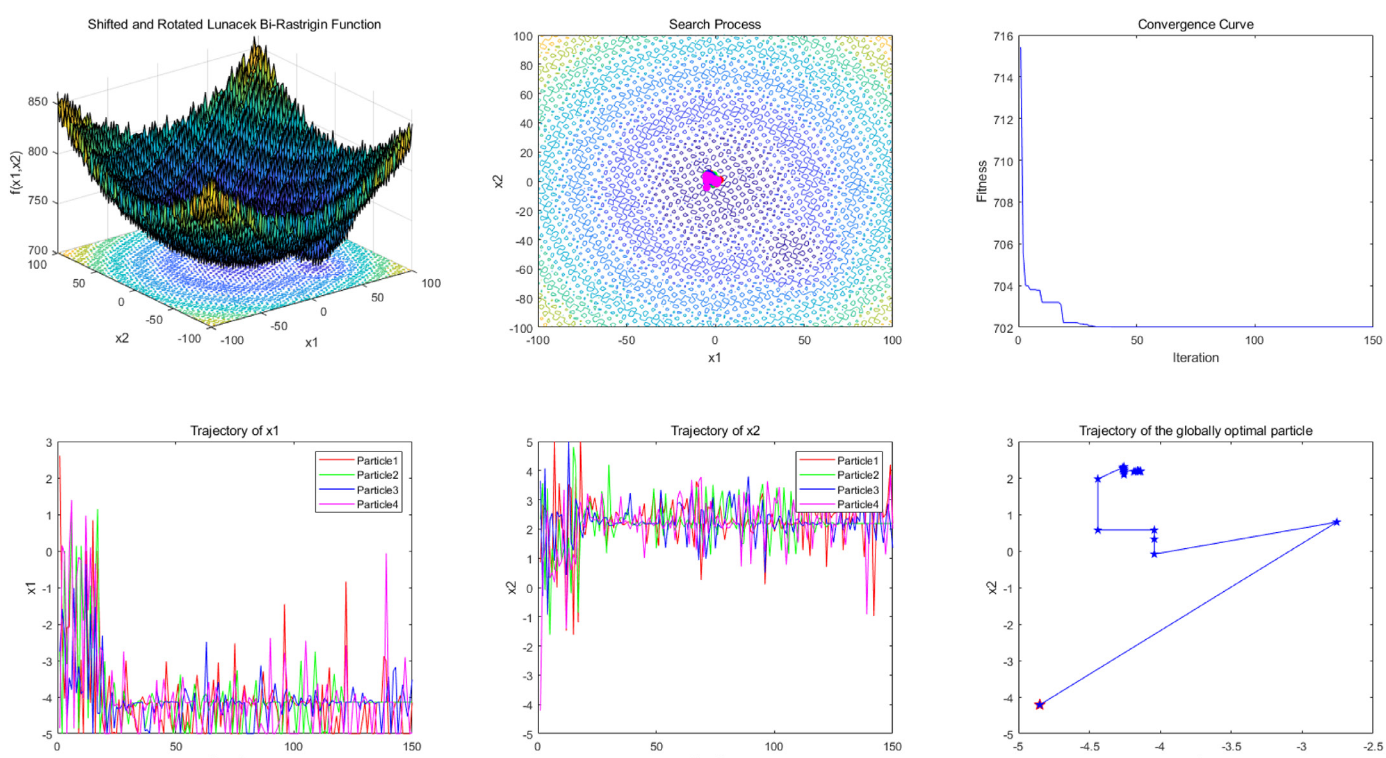



**Fig. 3.** Shifted and Rotated Lunacek Bi-Rastrigin function $f_7$.

$$= \left( \frac{1}{N} \sum_{i=1}^{N} Pbest_i^1(t), \frac{1}{N} \sum_{i=1}^{N} Pbest_i^2(t), \ldots, \frac{1}{N} \sum_{i=1}^{N} Pbest_i^D(t) \right) \tag{10}$$

where $N$ is the population size, $Pbest_i^d(t)(d = 1, 2, \ldots, D)$ is the $d^{th}$ component of the *Pbest* position of particle $i$ at iteration $t$.

In summary, stochastic and mainstream learning strategies have been proposed, and the original personal and global learning strategies will be replaced. Therefore, the velocity update equation will be changed as Eq. (11).

$$V_i(t+1) = \omega(t)V_i(t) + r_1c_1 \otimes (SPbest_i(t) - X_i(t))$$

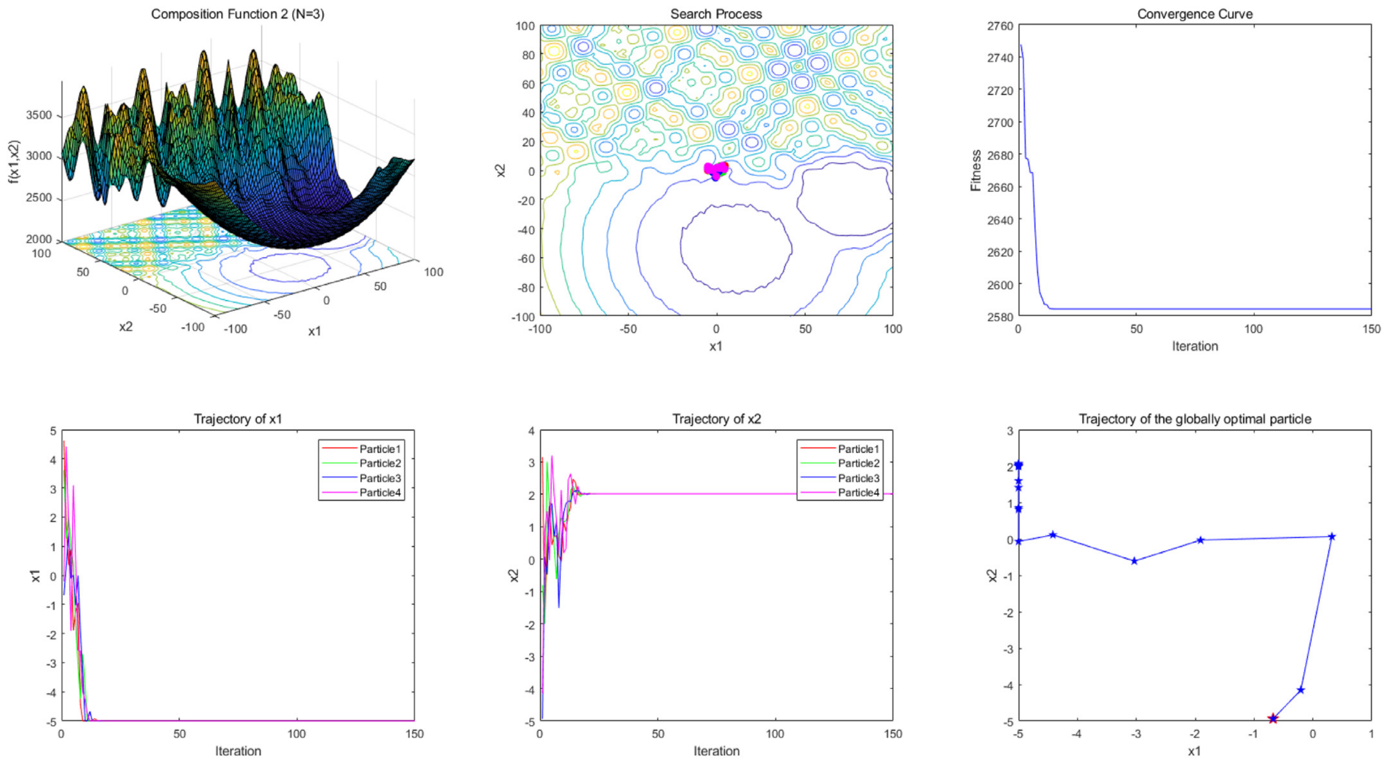



**Fig. 4.** Composition function 2 ($N = 3$) $f_{22}$.

$$+ r_2 c_2 \otimes (Mbest(t) - X_i(t)) \tag{11}$$

where $c_1 = c_2 = 2,\ r_1 \sim U(0,\ 1),\ r_2 \sim U(0,\ 1)$, and $\otimes$ denotes the element-wise product of two vectors. Similarly to the canonical PSO, on the right hand of Eq. (11), the first item is called "inertial component", the second item is called "stochastic learning component", and the third item is called "mainstream learning component".

### 3.3. Adaptive position updating strategy

It is well known that different position updating strategies have different exploration and exploitation capabilities. In our investigation, we found that "$X = X + V$" is beneficial to local exploitation, "$X = \omega X + (1 - \omega)V$" is conductive to the global exploration. In order to balance the local exploitation and global exploration, an adaptive position updating mechanism is proposed. Utilize this mechanism, particles can choose position updating strategies according to the corresponding conditions to balance exploration and exploitation better. As mentioned, the adaptive strategy position updating strategy is represented by Eqs. (12) and (13).

$$p_i = \frac{exp(fit(X_i(t)))}{exp\left(\frac{1}{N}\sum_{i=1}^{N} fit(X_i(t))\right)} \tag{12}$$

$$X_i(t+1) = \begin{cases} \omega(t)X_i(t) + (1-\omega(t))V_i(t+1) + Gbest(t) & p_i > rand \\ X_i(t) + V_i(t+1) & otherwise \end{cases} \tag{13}$$

where $fit(\ \cdot\ )$ is the fitness of the corresponding particle, $N$ is the size of population.

In the above adaptive strategy, the particles can obtain an estimate value at each iteration according to the ratio between the current fitness of particle $i$ and the average fitness of the population, which is recorded as $p_i$. when the $p_i$ is smaller, the performance of particle $i$ is higher than the average level of population, on this condition, in order to enhance its global exploration ability, the strategy of $X = X + V$ is adapted to update the position. On the contrary, when the $p_i$ is larger, particle $i$ is considered to be weaker than the average level of population, and the position updating strategy $X = \omega X + (1 - \omega)V$ is adopted to enhance its local exploitation ability.

### 3.4. Terminal replacement mechanism

In order to enhance the diversity of the population and improve the performance of MPSO on complex problems, terminal updating mechanism is introduced. Inspired by the survival of the fittest rule in nature, the global worst particle *Gworst* will be replaced at each iteration. Firstly, the global worst particle is defined as Eq. (14). Then, a new particle *Nbest* is generated by crossover as Eq. (15). Finally, comparing *Gworst* with *Nbest*, if the new generated *Nbest* is better than the *Gworst*, then the corresponding personal best of *Gworst* will be replaced by *Nbest*, otherwise, the corresponding personal best of *Gworst* will retain unchanged. This replacement mechanism is defined as Eq. (16).

$$Gworst(t) = argmax\{fit(Pbest_1(t)), fit(Pbest_2(t)), \cdots, fit(Pbest_N(t))\} \tag{14}$$

$$Nbest(t) = Gbest(t) + rand \cdot \left(Pbest_j(t) - Pbest_k(t)\right),\quad j \neq k \in \{1, 2, \cdots, N\} \tag{15}$$

$$Gworst(t) = \begin{cases} Nbest(t) & fit(Nbest(t)) < fit(Gworst(t)) \\ Gworst(t) & otherwise \end{cases} \tag{16}$$

In summary, MPSO as a variant of PSO is proposed, and the pseudocodes of implementing the proposed MPSO are listed as shown in Algorithm 1.

**Algorithm 1:** MPSO algorithm.

**Input**: Population size:$N$, the maximum number of iterations $T_{Max}$, inertia weight $\omega$, location boundary $[x^d_{\min}, x^d_{\max}]$, velocity boundary $[v^d_{\min}, v^d_{\max}]$ $d = 1, 2 \dots, D$, $v_{\max} = 0.5 \cdot (x_{\max} - x_{\min})$, $v_{\min} = -v_{\max}$ ;
**Output**: Optimal solution;

Initialize:$x^d_i = x^d_{\min} + rand \cdot (x^d_{\max} - x^d_{\min})$, $v^d_i = v^d_{\min} + rand \cdot (v^d_{\max} - v^d_{\min})$ ;
**for** $t = 1, 2, \dots, T_{Max}$ **do**
    Update $Ubest$ ;
    **for** $i = 1 : N$ **do**
        **if** $fit(Ubest) < fit(Pbest_i)$ **then**
            $Rbest_i = Ubest$ ;
        **end**
    **end**
    Update location of particle by Eqs. (12) and (13). Check the boundaries ;
    % Adaptive Position Updating Strategy ;
    **for** $i = 1 : N$ **do**
        **if** $p_i > rand$ **then**
            $X_i(t+1) = \omega(t) X_i(t) + (1 - \omega(t)) V_i(t+1) + Gbest$ ;
        **else**
            $X_i(t+1) = X_i(t) + V_i(t+1)$ ;
        **end**
    **end**
    Check the boundaries ;
    % Terminal Updating Strategy ;
    **for** $i = 1, 2, \dots, N$ **do**
        Update $Gworst_i$ by Eq. (14);
        Update $Nbest_i$ by Eq. (15) and Bounds checking.;
        **if** $fit(Nbest) < fit(Gworst)$ **then**
            $Gworst = Nbest$ ;
            $fit(Gworst) = fit(Nbest)$;
        **end**
    **end**
    **for** $i = 1, 2, \dots, N$ **do**
        **if** $fit(X_i) < fit(Pbest_i)$ **then**
            $Pbest_i = X_i$ ;
            $fit(Pbest_i) = fit(X_i)$
        **end**
        **if** $fit(Pbest_i) < fit(Gbest)$ **then**
            $Pbest_i = Gbest$;
            $fit(Gbest) = fit(Pbest_i)$;
        **end**
    **end**
**end**

## 4. Experimental results and discussion

### 4.1. Benchmark functions and comparison algorithms

In this experimentation, 30 benchmark functions in CEC2017 (Awad, Ali, Liang, Qu, & Suganthan, 2016) ($f_2$ has been excluded because it shows unstable behavior especially for higher dimensions, and significant performance variations for the same algorithm implemented in Matlab) are selected to evaluate the performance the proposed MPSO. These benchmark functions are classified into four categories: unimodal functions ($f_1 - f_3$), multimodal functions ($f_4 - f_{10}$), hybrid functions ($f_{11} - f_{20}$), composition functions ($f_{21} - f_{30}$), which are shown in Table 1 and Appendix.

To verify the performance of MPSO, 16 algorithm including 8 variations of PSO (PSO_cf (Eberhart & Shi, 2000), PSO_cf_local (Kennedy & Mendes, 2002), FIPS (Mendes, Kennedy, & Neves, 2004), CLPSO (Liang et al., 2006), DNLPSO (Nasir et al., 2012), FDR_PSO (Peram, Veeramachaneni, & Mohan, 2003), SPSO2011 (Clerc, 2011)) and 8 other evolution algorithms (ABC (Karaboga & Basturk, 2007), CSA (Yang & Deb, 2014), BOA (Arora & Singh, 2019), BSA (Civicioglu, 2013), DSA (Civicioglu, 2012), GSA (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009), OSA (Jain, Maurya, Rani, & Singh, 2018) and VSA (Doğan & Ölmez, 2015)) are evaluated to compared with the efficiency of MPSO. These evolution algorithms are listed as shown in Table 2.

For the sake of fairness, each algorithm is independently run 50 times, and the terminations condition of all algorithm is the maximum number of iterations, the maximum number of iteration is set 1000.

### 4.2. Results analysis and discussion

#### 4.2.1. Qualitative analysis of MPSO

For qualitative analysis and intuitive presentation of PSO iteration, four qualitative metrics are utilized. These metrics include search history, convergence curve, trajectory curve and movement of the global best particle. All qualitative metrics are carried out under the same conditions, in which the number of iterations of MPSO is set 150, and $D = 2$. We take $f_5, f_7, f_{22}$ and $f_{26}$ as examples to make qualitative analysis, which are shown in Figs. 2–5.

The trajectory curve represents the search process of particles in each dimension. The fluctuation of the previous indicates that the particles are searching exploring globally, while the stability of the late indicates that the particles have entered the global or local optimum.

#### 4.2.2. Comparisons between MPSO and 16 well-known algorithms

In this section, 8 variants of PSO and 8 other evolution algorithms are compared with MPSO, and the results are listed in Tables 3 and 4, respectively. Three indicators: average value(Mean), standard deviation(Std) and minimum(Min) are adopted. The convergence characteristics of all algorithms are shown in Figs. 6 and 7.

As can be seen from Table 3, the overall performance of MPSO are stronger than the 8 variants of PSO. From the comparison of mean value, the effect of MPSO is stronger than 8 variants of PSO on functions $f_3, f_{11}, f_{13}, f_{14}, f_{15}, f_{18}, f_{19}, f_{21}, f_{23}, f_{24}$ and $f_{30}$. From the minimum point of view, the performance of MPSO is stronger than the 8 variants of PSO on functions $f_3, f_7, f_9, f_{10}, f_{13}, f_{14}, f_{15}, f_{18}, f_{19}, f_{21}, f_{22}, f_{27}, f_{29}$ and $f_{30}$. In addition, MPSO does not perform well on some functions, such as $f_1, f_4, f_6, f_{17}$ and $f_{28}$. Although the performance of MPSO is unsatisfactory in some case, the performance of MPSO is still the best under the premise of ranking based on the mean of all functions.

From Table 4, we can see that the overall performance of MPSO is stronger than the 8 other evolution algorithm. It is clear that on functions $f_5, f_7, f_9, f_{11}, f_{12}, f_{13}, f_{20}, f_{21}, f_{23}, f_{24}, f_{29}$ and $f_{30}$, while on functions $f_3, f_{10}, f_{15}, f_{16}, f_{18}, f_{19}$ and $f_{26}$, MPSO is weaker than some other evolutionary algorithms. According to the comparison of the minimum values, the performance of MPSO is better than that of the other eight evolutionary algorithms on functions $f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{15}, f_{17}, f_{18}, f_{19}, f_{20}, f_{22}, f_{23}, f_{26}, f_{27}, f_{29}$ and $f_{30}$. Although MPSO performs poorly in some functions, it still performs best in the overall ranking.

#### 4.2.3. Statistical analysis

Generally, it is necessary to use statistical tests to analyse the experimental results. In this section, we focus on the use of nonparametric statistical inference for analysing the experimental results in Tables 3 and 4, which were obtained by MPSO and the other selected algorithms in the given search space.

**Table 1**
The information of the benchmark function used in this paper.

| NO. | Function | $D$ | Range | $f_{opt}$ |
|---|---|---|---|---|
| $f_1$ | Shifted and Rotated Bent Cigar Function | 30 | $[-100, 100]$ | 100 |
| $f_3$ | Shifted and Rotated Zakharov Function | 30 | $[-100, 100]$ | 300 |
| $f_4$ | Shifted and Rotated Rosenbrock's Function | 30 | $[-100, 100]$ | 400 |
| $f_5$ | Shifted and Rotated Rastrigin's Function | 30 | $[-100, 100]$ | 500 |
| $f_6$ | Shifted and Rotated Expanded Scaffer's F6 Function | 30 | $[-100, 100]$ | 600 |
| $f_7$ | Shifted and Rotated Lunacek Bi-Rastrigin Function | 30 | $[-100, 100]$ | 700 |
| $f_8$ | Shifted and Rotated Non-Continuous Rastrigin's Function | 30 | $[-100, 100]$ | 800 |
| $f_9$ | Shifted and Rotated Levy Function | 30 | $[-100, 100]$ | 900 |
| $f_{10}$ | Shifted and Rotated Schwefel's Function | 30 | $[-100, 100]$ | 1000 |
| $f_{11}$ | Hybrid Function 1 ($N = 3$) | 30 | $[-100, 100]$ | 1100 |
| $f_{12}$ | Hybrid Function 2 ($N = 3$) | 30 | $[-100, 100]$ | 1200 |
| $f_{13}$ | Hybrid Function 3 ($N = 3$) | 30 | $[-100, 100]$ | 1300 |
| $f_{14}$ | Hybrid Function 4 ($N = 4$) | 30 | $[-100, 100]$ | 1400 |
| $f_{15}$ | Hybrid Function 5 ($N = 4$) | 30 | $[-100, 100]$ | 1500 |
| $f_{16}$ | Hybrid Function 6 ($N = 4$) | 30 | $[-100, 100]$ | 1600 |
| $f_{17}$ | Hybrid Function 6 ($N = 5$) | 30 | $[-100, 100]$ | 1700 |
| $f_{18}$ | Hybrid Function 6 ($N = 5$) | 30 | $[-100, 100]$ | 1800 |
| $f_{19}$ | Hybrid Function 6 ($N = 5$) | 30 | $[-100, 100]$ | 1900 |
| $f_{20}$ | Hybrid Function 6 ($N = 6$) | 30 | $[-100, 100]$ | 2000 |
| $f_{21}$ | Composition Function 1 ($N = 3$) | 30 | $[-100, 100]$ | 2100 |
| $f_{22}$ | Composition Function 2 ($N = 3$) | 30 | $[-100, 100]$ | 2200 |
| $f_{23}$ | Composition Function 3 ($N = 4$) | 30 | $[-100, 100]$ | 2300 |
| $f_{24}$ | Composition Function 4 ($N = 4$) | 30 | $[-100, 100]$ | 2400 |
| $f_{25}$ | Composition Function 5 ($N = 5$) | 30 | $[-100, 100]$ | 2500 |
| $f_{26}$ | Composition Function 6 ($N = 5$) | 30 | $[-100, 100]$ | 2600 |
| $f_{27}$ | Composition Function 7 ($N = 6$) | 30 | $[-100, 100]$ | 2700 |
| $f_{28}$ | Composition Function 8 ($N = 6$) | 30 | $[-100, 100]$ | 2800 |
| $f_{29}$ | Composition Function 9 ($N = 3$) | 30 | $[-100, 100]$ | 2900 |
| $f_{30}$ | Composition Function 10 ($N = 3$) | 30 | $[-100, 100]$ | 3000 |

**Table 2**
Some well-known variants of PSO and other evolution algorithms.

| Algorithm | Years | Parameter information |
|---|---|---|
| PSO (Shi & Eberhart, 1998) | 1998 | $\omega = 0.9 \sim 0.4,\ c_1 = c_2 = 2.$ |
| PSO_cf (Eberhart & Shi, 2000) | 2000 | $\omega = 0.9 \sim 0.4,\ c_1 = c_2 = 2$ |
| PSO_cf_local (Kennedy & Mendes, 2002) | 2002 | $\omega = 0.9 \sim 0.4,\ c_1 = c_2 = 2$ |
| FIPS (Mendes et al., 2004) | 2004 | $\chi = 0.729,\ \Sigma c_i = 4.1.$ |
| FDR_PSO (Peram et al., 2003) | 2003 | $\omega = 0.4 - 0.9, c_1 = c_2 = 1, c_3 = 2$ |
| CLPSO (Liang et al., 2006) | 2006 | $w = 0.9 \sim 0.4,\ c = 1.49445,\ m = 7.$ |
| DNLPSO (Nasir et al., 2012) | 2012 | $\omega = 0.9 \sim 0.4,\ c_1 + c_2 = 2,\ \delta$: $randomin[0.05, 0.1]$, $\sigma = 1 \sim 0.1.$ |
| SPSO2011 (Clerc, 2011) | 2011 | $\omega = 0.9 \sim 0.4,\ c_1 = c_2 = 2.$ |
| ABC (Karaboga & Basturk, 2007) | 2007 | $limit = 100,\ Size\ of\ employed - bee = N/2$ |
| GSA (Rashedi et al., 2009) | 2009 | $\alpha = 20,\ G_0 = 100$ |
| CSA (Yang & Deb, 2014) | 2009 | Discovery rate of alien eggs: $pa = 0.25$ |
| DSA (Civicioglu, 2012) | 2012 | $method = 1,\ p1 = 0.3rand,\ p2 = 0.3rand,\ R = 1./gamrnd(1, 0.5)$ |
| BSA (Civicioglu, 2013) | 2013 | $mixrate = 1$ |
| VSA (Doğan & Ölmez, 2015) | 2015 | $x = 0.1,\ ginv = (1/x) * gammaincinv(x, 1)$ |
| OSA (Jain et al., 2018) | 2018 | $\beta = 1.9 \sim 0$ |
| BOA (Arora & Singh, 2019) | 2019 | $Sensormodality\ c = 0 \sim 1,\ Powerexponent$: 0.1 ~ 0.3, $Switchprobability$: $p = 0 \sim 8$ |
| MPSO | presented | |

To draw a statistical conclusion, a pair-wise statistical test named Wilcoxon Signed-Rank Test (WSRT) is used to better compare the overall performance of the algorithms. WSRT is a non-parametric test that can be used to check for the statistical significance difference between two algorithms. The null hypothesis $H_0$ for a two-sided test is:" there is no difference between the median of the solutions produced by algorithm A and the median of the solutions produced by algorithm B for the same benchmark function" (Zhang, Li, Li, Lai, & Zhang, 2016). To determine whether algorithm A achieves a statistically better solution than algorithm B, or if not, whether the alternative hypothesis is valid, the sizes of the ranks provided by WSRT are examined. When using WSRT, the R+ and R− related to the comparisons between two algorithms can be calculated and their $p$-values can be obtained. In this part, WSRT is used for statistical analysis with the significance level $\alpha = 0.05$. For the statistical analysis, the best solutions of 50 runs for each benchmark function are used for their pair-wise comparisons.

For MPSO and the variants of PSO, the statistical comparisons of the algorithms by WSRT are shown in Tables 5 and 6. In this tables, '+' indicates the case in which the null hypothesis is rejected and MPSO shows a better performance in the statistical comparison tests at 95% significance level ($\alpha = 0.05$); '−' indicates the case in which the null hypothesis is rejected and MPSO shows a worse performance; and '=' indicates a case in which no statistically significant difference between MPSO and the other algorithms exists. The last rows of Tables 5 and 6 show the total counts in the '+/=/−' format. From Tables 5 and 6, we can conclude that MPSO performs significantly better than any other algorithms on major benchmark functions.

For MPSO and the other evolutionary algorithms, the statistical comparisons of the algorithms by WSRT are shown in Tables 7 and 8. These statistical results show that MPSO is evidently superior to all other algorithms.

**Table 3**
Comparisons of experimental results between MPSO with some well-known variants of PSO.

| | | SPSO | PSO_cf | FIPS | FDR_PSO | CLPSO | SPSO2011 | DNLPSO | PSO_cf_local | MPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $f_1$ | Main | 1.42E+10 | 8.30E+09 | 4.41E+06 | 1.57E+08 | 4.68E+07 | **2.82E+03** | 8.74E+09 | 6.96E+07 | 8.19E+07 |
| | Std | 6.66E+09 | 5.71E+09 | 1.52E+06 | 4.49E+08 | 1.03E+07 | **2.18E+03** | 5.85E+09 | 2.61E+08 | 1.80E+08 |
| | Min | 4.12E+09 | 2.09E+04 | 1.86E+06 | 1.02E+02 | 2.66E+07 | **1.10E+02** | 2.09E+04 | 1.00E+02 | 1.71E+03 |
| | Rank | 9 | 7 | 2 | 6 | 3 | **1** | 8 | 4 | 5 |
| $f_3$ | Main | 8.52E+04 | 4.52E+04 | 7.33E+04 | 1.59E+04 | 9.17E+04 | 3.09E+04 | 6.68E+04 | 6.80E+04 | **1.15E+04** |
| | Std | 3.54E+04 | 2.46E+04 | 1.38E+04 | **6.35E+03** | 1.46E+04 | 8.78E+03 | 3.32E+04 | 1.97E+04 | 7.04E+03 |
| | Min | 2.75E+04 | 1.26E+04 | 4.41E+04 | 5.59E+03 | 5.59E+04 | 1.63E+04 | 1.76E+04 | 2.73E+04 | **3.51E+03** |
| | Rank | 8 | 4 | 7 | 2 | 9 | 3 | 5 | 6 | **1** |
| $f_4$ | Main | 1.41E+03 | 1.19E+03 | 4.99E+02 | 5.00E+02 | 5.65E+02 | **4.81E+02** | 1.70E+03 | 5.04E+02 | 5.04E+02 |
| | Std | 8.72E+02 | 9.07E+02 | **7.67E+00** | 2.88E+01 | 1.04E+01 | 3.67E+01 | 1.32E+03 | 2.26E+01 | 2.56E+01 |
| | Min | 5.70E+02 | 5.04E+02 | 4.90E+02 | 4.07E+02 | 5.42E+02 | **4.00E+02** | 5.21E+02 | 4.87E+02 | 4.23E+02 |
| | Rank | 8 | 7 | 2 | 3 | 6 | **1** | 9 | 4 | 5 |
| $f_5$ | Main | 6.48E+02 | 6.44E+02 | 6.80E+02 | **5.63E+02** | 6.57E+02 | 6.26E+02 | 6.17E+02 | 5.66E+02 | 5.70E+02 |
| | Std | 3.01E+01 | 3.82E+01 | 1.46E+01 | 1.86E+01 | **1.05E+01** | 1.93E+01 | 3.44E+01 | 1.76E+01 | 1.75E+01 |
| | Min | 5.92E+02 | 5.63E+02 | 6.42E+02 | 5.36E+02 | 6.27E+02 | 5.78E+02 | 5.57E+02 | **5.26E+02** | 5.32E+02 |
| | Rank | 7 | 6 | 9 | **1** | 8 | 5 | 4 | 2 | 3 |
| $f_6$ | Main | 6.20E+02 | 6.16E+02 | 6.01E+02 | 6.01E+02 | 6.04E+02 | 6.04E+02 | 6.11E+02 | **6.00E+02** | 6.05E+02 |
| | Std | 9.83E+00 | 8.09E+00 | 2.13E−01 | 8.36E−01 | 4.99E−01 | 1.89E+00 | 7.62E+00 | **4.00E−01** | 3.72E+00 |
| | Min | 6.04E+02 | 6.05E+02 | 6.01E+02 | **6.00E+02** | 6.03E+02 | 6.01E+02 | 6.02E+02 | **6.00E+02** | 6.01E+02 |
| | Rank | 9 | 8 | 3 | 2 | 4 | 5 | 7 | **1** | 6 |
| $f_7$ | Main | 9.57E+02 | 8.93E+02 | 9.27E+02 | 7.98E+02 | 9.53E+02 | 8.61E+02 | 8.70E+02 | **7.94E+02** | 8.08E+02 |
| | Std | 1.32E+02 | 7.89E+01 | 1.36E+01 | 2.22E+01 | 1.64E+01 | 2.00E+01 | 7.72E+01 | **1.60E+01** | 2.17E+01 |
| | Min | 7.93E+02 | 7.90E+02 | 8.90E+02 | 7.51E+02 | 9.01E+02 | 7.93E+02 | 7.63E+02 | 7.67E+02 | **7.56E+02** |
| | Rank | 9 | 6 | 7 | 2 | 8 | 4 | 5 | **1** | 3 |
| $f_8$ | Main | 9.47E+02 | 9.41E+02 | 9.87E+02 | **8.60E+02** | 9.60E+02 | 9.19E+02 | 9.02E+02 | 8.68E+02 | 8.68E+02 |
| | Std | 3.17E+01 | 4.19E+01 | **1.20E+01** | 1.65E+01 | 1.26E+01 | 1.99E+01 | 2.77E+01 | 1.58E+01 | 2.17E+01 |
| | Min | 8.65E+02 | 8.76E+02 | 9.57E+02 | **8.31E+02** | 9.32E+02 | 8.57E+02 | 8.57E+02 | 8.41E+02 | 8.33E+02 |
| | Rank | 7 | 6 | 9 | **1** | 8 | 5 | 4 | 2 | 3 |
| $f_9$ | Main | 5.04E+03 | 3.08E+03 | **9.19E+02** | 9.50E+02 | 2.57E+03 | 9.46E+02 | 3.63E+03 | 9.42E+02 | 1.05E+03 |
| | Std | 2.17E+03 | 1.50E+03 | **7.06E+00** | 1.11E+02 | 4.40E+02 | 3.93E+01 | 1.91E+03 | 6.89E+01 | 1.82E+02 |
| | Min | 1.37E+03 | 1.28E+03 | 9.09E+02 | 9.01E+02 | 1.58E+03 | 9.02E+02 | 9.84E+02 | 9.02E+02 | **9.07E+02** |
| | Rank | 9 | 7 | 1 | 4 | 6 | 3 | 8 | 2 | 5 |
| $f_{10}$ | Main | 5.30E+03 | 4.71E+03 | 7.82E+03 | 4.47E+03 | 6.26E+03 | 7.22E+03 | 6.04E+03 | **4.43E+03** | 4.49E+03 |
| | Std | 7.56E+02 | 8.36E+02 | 3.59E+02 | 7.00E+02 | 2.88E+02 | 4.88E+02 | 1.27E+03 | **5.36E+02** | 8.35E+02 |
| | Min | 4.05E+03 | 3.10E+03 | 7.03E+03 | 2.97E+03 | 5.62E+03 | 6.18E+03 | 3.33E+03 | 3.21E+03 | **2.74E+03** |
| | Rank | 5 | 4 | 9 | 2 | 7 | 8 | 6 | **1** | 3 |
| $f_{11}$ | Main | 1.77E+03 | 1.37E+03 | 1.26E+03 | 1.22E+03 | 1.44E+03 | 1.23E+03 | 1.55E+03 | 1.24E+03 | **1.22E+03** |
| | Std | 6.63E+02 | 1.06E+02 | 1.80E+01 | 4.01E+01 | 6.20E+01 | 4.43E+01 | 7.46E+02 | 4.57E+01 | **5.02E+01** |
| | Min | 1.32E+03 | 1.18E+03 | 1.22E+03 | 1.14E+03 | 1.31E+03 | 1.15E+03 | 1.24E+03 | **1.13E+03** | 1.14E+03 |
| | Rank | 9 | 6 | 5 | 2 | 7 | 3 | 8 | 4 | **1** |
| $f_{12}$ | Main | 9.95E+08 | 4.50E+08 | 2.90E+06 | 2.27E+06 | 1.16E+07 | **1.02E+06** | 6.47E+08 | 8.50E+06 | 1.51E+06 |
| | Std | 1.19E+09 | 6.28E+08 | 1.16E+06 | 5.53E+06 | 3.54E+06 | **6.32E+05** | 9.17E+08 | 2.85E+07 | 1.37E+06 |
| | Min | 2.14E+07 | 8.82E+05 | 1.09E+06 | **4.48E+04** | 5.21E+06 | 1.19E+05 | 1.66E+05 | 8.52E+04 | 4.83E+04 |
| | Rank | 9 | 7 | 4 | 3 | 6 | **1** | 8 | 5 | 2 |
| $f_{13}$ | Main | 7.00E+08 | 3.61E+08 | 5.68E+04 | 1.20E+05 | 1.03E+06 | 4.12E+04 | 1.69E+08 | 2.09E+06 | **1.56E+04** |
| | Std | 1.17E+09 | 7.05E+08 | 4.48E+04 | 6.26E+05 | 6.51E+05 | 1.89E+04 | 4.26E+08 | 1.02E+07 | **1.11E+04** |
| | Min | 9.27E+04 | 1.40E+04 | 4.33E+03 | 2.68E+03 | 2.45E+05 | 8.59E+03 | 3.83E+04 | 6.42E+03 | **1.56E+03** |
| | Rank | 9 | 8 | 3 | 4 | 5 | 2 | 7 | 6 | **1** |
| $f_{14}$ | Main | 3.21E+05 | 9.17E+04 | 8.17E+04 | 3.50E+04 | 6.13E+04 | 3.49E+04 | 2.48E+05 | 4.69E+04 | **5.00E+03** |
| | Std | 1.01E+06 | 8.57E+04 | 5.93E+04 | 4.03E+04 | 4.41E+04 | 2.55E+04 | 6.28E+05 | 4.68E+04 | **4.87E+03** |
| | Min | 1.37E+04 | 7.03E+03 | 4.01E+03 | 1.84E+03 | 7.92E+03 | 5.35E+03 | 1.01E+04 | 2.85E+03 | **1.58E+03** |
| | Rank | 9 | 7 | 6 | 3 | 5 | 2 | 8 | 4 | **1** |
| $f_{15}$ | Main | 1.34E+05 | 1.81E+07 | 1.97E+04 | 8.97E+03 | 3.72E+04 | 6.60E+04 | 6.79E+04 | 1.73E+04 | **3.40E+03** |
| | Std | 9.97E+04 | 1.28E+08 | 1.55E+04 | 1.01E+04 | 1.84E+04 | 6.15E+04 | 4.55E+04 | 1.80E+04 | **2.40E+03** |
| | Min | 7.27E+03 | 2.86E+03 | 2.33E+03 | 1.67E+03 | 9.37E+03 | 1.03E+04 | 2.35E+03 | 2.29E+03 | **1.60E+03** |
| | Rank | 8 | 9 | 4 | 2 | 5 | 6 | 7 | 3 | **1** |
| $f_{16}$ | Main | 2.96E+03 | 2.91E+03 | 2.90E+03 | 2.44E+03 | 2.47E+03 | 2.67E+03 | 2.94E+03 | **2.37E+03** | 2.49E+03 |
| | Std | 3.81E+02 | 3.57E+02 | 2.74E+02 | 2.46E+02 | **1.60E+02** | 2.68E+02 | 4.30E+02 | 2.69E+02 | 2.46E+02 |
| | Min | 2.03E+03 | 1.98E+03 | 2.21E+03 | 1.86E+03 | 2.12E+03 | 2.12E+03 | 2.05E+03 | **1.76E+03** | 1.98E+03 |
| | Rank | 9 | 7 | 6 | 2 | 3 | 5 | 8 | **1** | 4 |
| $f_{17}$ | Main | 2.44E+03 | 2.46E+03 | **1.94E+03** | 1.99E+03 | 1.94E+03 | 2.00E+03 | 2.38E+03 | 1.99E+03 | 2.05E+03 |
| | Std | 2.82E+02 | 3.00E+02 | **7.76E+01** | 1.35E+02 | 7.10E+01 | 1.02E+02 | 3.07E+02 | 1.17E+02 | 1.76E+02 |
| | Min | 1.86E+03 | 1.93E+03 | 1.81E+03 | **1.75E+03** | 1.80E+03 | 1.86E+03 | 1.83E+03 | 1.82E+03 | 1.77E+03 |
| | Rank | 8 | 9 | **1** | 4 | 2 | 5 | 7 | 3 | 6 |
| $f_{18}$ | Main | 2.97E+06 | 1.70E+06 | 7.55E+05 | 3.00E+05 | 5.42E+05 | 5.57E+05 | 2.09E+06 | 4.42E+05 | **1.20E+05** |
| | Std | 5.08E+06 | 3.21E+06 | 3.39E+05 | 1.96E+05 | 2.93E+05 | 5.68E+05 | 2.63E+06 | 4.48E+05 | **1.19E+05** |
| | Min | 7.20E+04 | 3.70E+04 | 2.00E+05 | 5.88E+04 | 4.28E+04 | 5.72E+04 | 1.25E+05 | 4.66E+04 | **2.74E+04** |
| | Rank | 9 | 7 | 6 | 2 | 4 | 5 | 8 | 3 | **1** |
| $f_{19}$ | Main | 1.71E+07 | 7.40E+06 | 9.04E+03 | 1.40E+04 | 3.10E+04 | 7.13E+05 | 1.67E+07 | 1.98E+05 | **6.33E+03** |
| | Std | 4.18E+07 | 2.89E+07 | 6.99E+03 | 2.66E+04 | 1.53E+04 | 6.38E+05 | 4.10E+07 | 1.09E+06 | **4.51E+03** |
| | Min | 5.77E+03 | 1.94E+03 | 2.35E+03 | 1.94E+03 | 4.06E+03 | 8.85E+04 | 3.05E+03 | 2.11E+03 | **1.92E+03** |
| | Rank | 9 | 7 | 2 | 3 | 4 | 6 | 8 | 5 | **1** |
| $f_{20}$ | Main | 2.41E+03 | 2.51E+03 | 2.42E+03 | 2.30E+03 | 2.30E+03 | 2.40E+03 | 2.37E+03 | **2.27E+03** | 2.30E+03 |
| | Std | 1.71E+02 | 1.63E+02 | 9.84E+01 | 1.36E+02 | 8.08E+01 | **1.02E+02** | 1.74E+02 | 1.26E+02 | 1.37E+02 |
| | Min | 2.14E+03 | 2.13E+03 | 2.22E+03 | 2.05E+03 | 2.14E+03 | 2.18E+03 | 2.08E+03 | **2.05E+03** | 2.06E+03 |
| | Rank | 7 | 9 | 8 | 3 | 4 | 6 | 5 | **1** | 2 |

(*continued on next page*)

**Table 3** (*continued*)

| | | SPSO | PSO_cf | FIPS | FDR_PSO | CLPSO | SPSO2011 | DNLPSO | PSO_cf_local | MPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $f_{21}$ | Main | 2.45E+03 | 2.44E+03 | 2.48E+03 | 2.36E+03 | 2.44E+03 | 2.41E+03 | 2.43E+03 | 2.37E+03 | **2.36E+03** |
| | Std | 3.65E+01 | 4.20E+01 | **1.17E+01** | 1.59E+01 | 3.70E+01 | 1.76E+01 | 3.15E+01 | 1.73E+01 | 1.71E+01 |
| | Min | 2.37E+03 | 2.36E+03 | 2.45E+03 | 2.34E+03 | 2.29E+03 | 2.37E+03 | 2.38E+03 | 2.34E+03 | **2.33E+03** |
| | Rank | 8 | 7 | 9 | 2 | 6 | 4 | 5 | 3 | **1** |
| $f_{22}$ | Main | 6.01E+03 | 5.17E+03 | 2.32E+03 | 3.52E+03 | 2.94E+03 | **2.30E+03** | 6.34E+03 | 3.42E+03 | 2.79E+03 |
| | Std | 1.40E+03 | 1.50E+03 | 9.73E−01 | 1.71E+03 | 1.07E+03 | **1.13E+00** | 1.96E+03 | 1.58E+03 | 1.26E+03 |
| | Min | 3.00E+03 | 2.59E+03 | 2.31E+03 | 2.30E+03 | 2.45E+03 | **2.30E+03** | 2.46E+03 | **2.30E+03** | **2.30E+03** |
| | Rank | 8 | 7 | 2 | 6 | 4 | **1** | 9 | 5 | 3 |
| $f_{23}$ | Main | 2.92E+03 | 2.92E+03 | 2.83E+03 | 2.79E+03 | 2.79E+03 | 2.76E+03 | 2.90E+03 | 2.77E+03 | **2.73E+03** |
| | Std | 6.24E+01 | 8.88E+01 | 1.35E+01 | 4.90E+01 | **1.10E+01** | 3.56E+01 | 7.06E+01 | 2.22E+01 | 3.42E+01 |
| | Min | 2.79E+03 | 2.78E+03 | 2.78E+03 | 2.73E+03 | 2.76E+03 | **2.66E+03** | 2.78E+03 | 2.73E+03 | 2.67E+03 |
| | Rank | 8 | 9 | 6 | 5 | 4 | 2 | 7 | 3 | **1** |
| $f_{24}$ | Main | 3.13E+03 | 3.10E+03 | 3.01E+03 | 2.99E+03 | 2.98E+03 | 2.91E+03 | 3.10E+03 | 2.94E+03 | **2.90E+03** |
| | Std | 7.33E+01 | 6.16E+01 | **1.53E+01** | 5.21E+01 | 1.67E+01 | 3.67E+01 | 6.59E+01 | 3.10E+01 | 3.34E+01 |
| | Min | 3.01E+03 | 2.99E+03 | 2.98E+03 | 2.90E+03 | 2.94E+03 | **2.84E+03** | 2.98E+03 | 2.88E+03 | 2.85E+03 |
| | Rank | 9 | 7 | 6 | 5 | 4 | 2 | 8 | 3 | **1** |
| $f_{25}$ | Main | 3.22E+03 | 3.09E+03 | **2.89E+03** | 2.89E+03 | 2.93E+03 | 2.92E+03 | 3.06E+03 | 2.89E+03 | 2.95E+03 |
| | Std | 2.58E+02 | 2.79E+02 | **1.49E+00** | 1.42E+01 | 9.74E+00 | 2.34E+01 | 1.51E+02 | 1.42E+01 | 3.33E+01 |
| | Min | 2.92E+03 | 2.89E+03 | 2.89E+03 | **2.88E+03** | 2.91E+03 | 2.89E+03 | 2.89E+03 | 2.89E+03 | 2.89E+03 |
| | Rank | 9 | 8 | **1** | 3 | 5 | 4 | 7 | 2 | 6 |
| $f_{26}$ | Main | 6.45E+03 | 6.34E+03 | **3.01E+03** | 4.33E+03 | 4.81E+03 | 4.51E+03 | 6.32E+03 | 4.79E+03 | 4.43E+03 |
| | Std | 7.17E+02 | 8.42E+02 | 3.51E+02 | 6.10E+02 | 4.67E+02 | 5.68E+02 | 7.86E+02 | 2.94E+02 | 1.31E+03 |
| | Min | 5.03E+03 | 4.10E+03 | 2.86E+03 | 2.80E+03 | 3.83E+03 | **2.80E+03** | 4.72E+03 | 4.25E+03 | 2.81E+03 |
| | Rank | 9 | 8 | **1** | 2 | 6 | 4 | 7 | 5 | 3 |
| $f_{27}$ | Main | 3.33E+03 | 3.34E+03 | **3.21E+03** | 3.25E+03 | 3.24E+03 | 3.25E+03 | 3.33E+03 | 3.24E+03 | 3.24E+03 |
| | Std | 6.05E+01 | 6.86E+01 | **6.44E+00** | 2.21E+01 | 5.05E+00 | 2.01E+01 | 6.73E+01 | 2.55E+01 | 1.59E+01 |
| | Min | 3.23E+03 | 3.21E+03 | **3.20E+03** | 3.21E+03 | 3.23E+03 | 3.22E+03 | 3.23E+03 | **3.20E+03** | **3.20E+03** |
| | Rank | 7 | 9 | **1** | 6 | 3 | 5 | 8 | 2 | 4 |
| $f_{28}$ | Main | 4.27E+03 | 3.90E+03 | 3.25E+03 | 3.26E+03 | 3.33E+03 | **3.16E+03** | 4.33E+03 | 3.43E+03 | 3.35E+03 |
| | Std | 7.91E+02 | 7.05E+02 | **1.16E+01** | 4.46E+01 | 1.69E+01 | 5.18E+01 | 1.21E+03 | 4.97E+02 | 9.10E+01 |
| | Min | 3.39E+03 | 3.28E+03 | 3.22E+03 | 3.21E+03 | 3.30E+03 | **3.10E+03** | 3.29E+03 | 3.22E+03 | 3.26E+03 |
| | Rank | 8 | 7 | 2 | 3 | 4 | **1** | 9 | 6 | 5 |
| $f_{29}$ | Main | 4.12E+03 | 4.15E+03 | 3.99E+03 | **3.65E+03** | 3.74E+03 | 3.78E+03 | 4.08E+03 | 3.80E+03 | 3.67E+03 |
| | Std | 3.73E+02 | 3.23E+02 | **1.12E+02** | 1.78E+02 | 9.75E+01 | 1.18E+02 | 3.87E+02 | 1.97E+02 | 1.89E+02 |
| | Min | 3.59E+03 | 3.54E+03 | 3.70E+03 | 3.37E+03 | 3.59E+03 | 3.58E+03 | 3.49E+03 | 3.47E+03 | **3.36E+03** |
| | Rank | 8 | 9 | 6 | **1** | 3 | 4 | 7 | 5 | 2 |
| $f_{30}$ | Main | 5.47E+06 | 3.40E+06 | 1.95E+05 | 3.71E+04 | 2.68E+05 | 9.39E+05 | 6.30E+06 | 1.51E+05 | **1.21E+04** |
| | Std | 5.57E+06 | 5.55E+06 | 1.40E+05 | 8.96E+04 | 1.29E+05 | 7.26E+05 | 1.63E+07 | 4.04E+05 | **1.20E+04** |
| | Min | 7.48E+04 | 8.47E+03 | 3.57E+04 | 5.91E+03 | 6.72E+04 | 1.62E+05 | 1.87E+04 | 6.42E+03 | **5.60E+03** |
| | Rank | 8 | 7 | 4 | 2 | 5 | 6 | 9 | 3 | **1** |
| ToTal Rank | | 239 | 209 | 132 | 86 | 148 | 109 | 206 | 95 | 81 |
| Final Rank | | 9 | 8 | 5 | 2 | 6 | 4 | 7 | 3 | 1 |

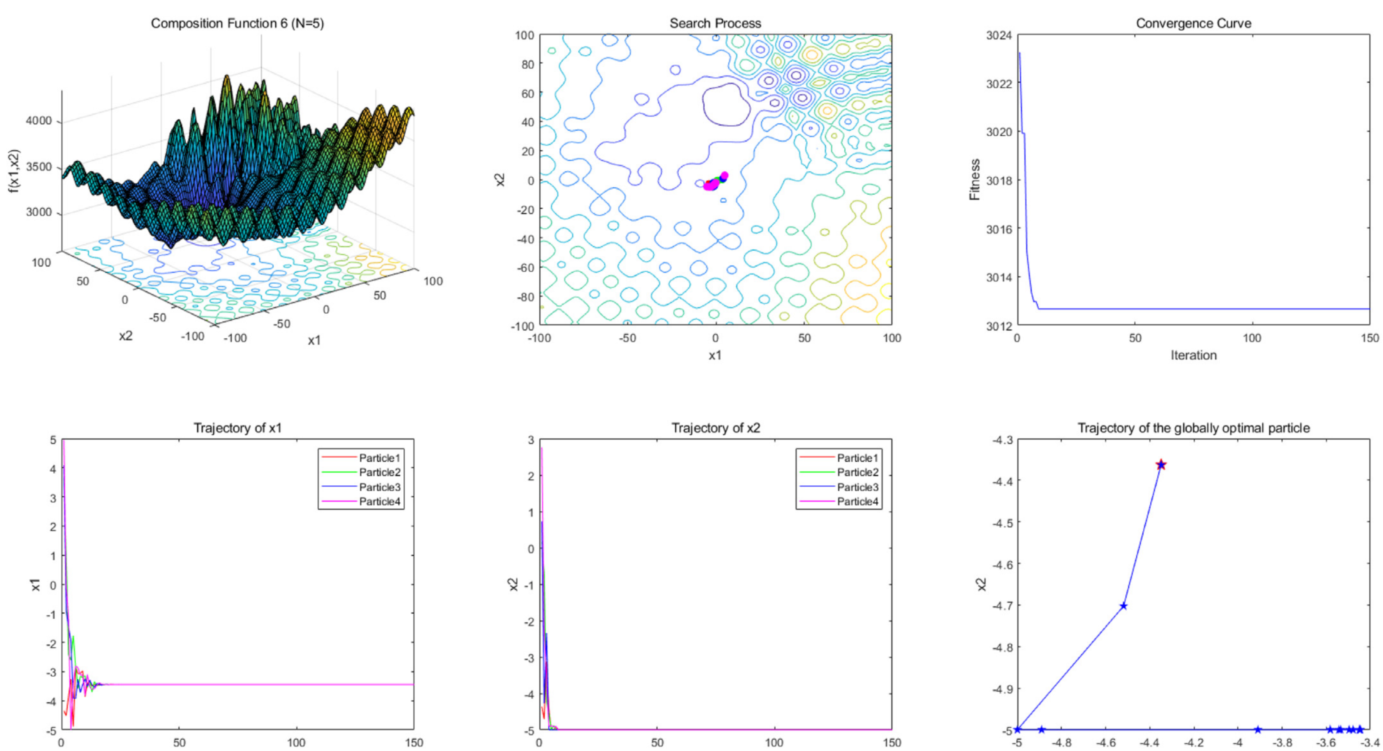



**Fig. 5.** Composition function 6 ($N = 5$) $f_{26}$.

**Table 4**
Comparisons of experimental results between MPSO with 8 evolutionary algorithms.

| | | ABC | VSA | CSA | BOA | BSA | DSA | OSA | GSA | MPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $f_1$ | Mean | **2.70E+03** | 4.64E+03 | 1.00E+10 | 5.05E+10 | 9.71E+04 | 2.10E+05 | 5.31E+10 | 1.03E+08 | 8.19E+07 |
| | Std | 2.31E+03 | 5.41E+03 | **0.00E+00** | 7.42E+09 | 6.68E+04 | 1.48E+05 | 6.69E+09 | 1.49E+08 | 1.80E+08 |
| | Min | 1.83E+02 | **1.00E+02** | 1.00E+10 | 3.71E+10 | 1.56E+04 | 3.46E+04 | 4.08E+10 | 3.91E+02 | 1.71E+03 |
| | Rank | **1** | 2 | 7 | 8 | 3 | 4 | 9 | 6 | 5 |
| $f_3$ | Mean | 1.31E+05 | **4.40E+03** | 1.00E+05 | 8.43E+04 | 6.41E+04 | 1.13E+05 | 9.23E+04 | 9.14E+04 | 1.15E+04 |
| | Std | 1.87E+04 | 2.61E+03 | 1.93E+04 | 6.38E+03 | 9.05E+03 | 2.37E+04 | **2.19E+03** | 8.46E+03 | 7.04E+03 |
| | Min | 7.26E+04 | **9.96E+02** | 6.36E+04 | 6.33E+04 | 4.76E+04 | 5.82E+04 | 8.28E+04 | 7.17E+04 | 3.51E+03 |
| | Rank | 9 | **1** | 7 | 4 | 3 | 8 | 6 | 5 | 2 |
| $f_4$ | Mean | **4.71E+02** | 5.02E+02 | 4.92E+02 | 2.01E+04 | 5.19E+02 | 5.23E+02 | 1.21E+04 | 6.89E+02 | 5.04E+02 |
| | Std | 1.76E+01 | 1.49E+01 | 1.82E+01 | 2.88E+03 | 9.18E+00 | **1.34E+01** | 2.34E+03 | 1.40E+02 | 2.56E+01 |
| | Min | **4.20E+02** | 4.64E+02 | 4.39E+02 | 1.43E+04 | 4.89E+02 | 4.92E+02 | 7.28E+03 | 5.43E+02 | 4.23E+02 |
| | Rank | **1** | 3 | 2 | 9 | 5 | 6 | 8 | 7 | 4 |
| $f_5$ | Mean | 6.03E+02 | 6.09E+02 | 6.72E+02 | 9.05E+02 | 6.14E+02 | 6.32E+02 | 9.45E+02 | 7.37E+02 | **5.70E+02** |
| | Std | 1.89E+01 | 3.16E+01 | 2.85E+01 | 2.20E+01 | **1.09E+01** | 1.18E+01 | 2.30E+01 | 2.49E+01 | 1.75E+01 |
| | Min | 5.62E+02 | 5.63E+02 | 6.11E+02 | 8.40E+02 | 5.89E+02 | 6.07E+02 | 8.97E+02 | 6.83E+02 | **5.32E+02** |
| | Rank | 2 | 3 | 6 | 8 | 4 | 5 | 9 | 7 | **1** |
| $f_6$ | Mean | **6.00E+02** | 6.22E+02 | 6.55E+02 | 6.85E+02 | 6.01E+02 | 6.00E+02 | 6.95E+02 | 6.58E+02 | 6.05E+02 |
| | Std | **1.33E−02** | 1.03E+01 | 9.22E+00 | 5.29E+00 | 4.52E−01 | 1.30E−01 | 5.54E+00 | 4.51E+00 | 3.72E+00 |
| | Min | **6.00E+02** | 6.07E+02 | 6.34E+02 | 6.74E+02 | **6.00E+02** | **6.00E+02** | 6.82E+02 | 6.50E+02 | 6.01E+02 |
| | Rank | **1** | 5 | 6 | 8 | 3 | 2 | 9 | 7 | 4 |
| $f_7$ | Mean | 8.29E+02 | 8.63E+02 | 9.37E+02 | 1.38E+03 | 8.56E+02 | 8.73E+02 | 1.46E+03 | 9.56E+02 | **8.08E+02** |
| | Std | 1.31E+01 | 4.29E+01 | 3.29E+01 | 3.40E+01 | 1.41E+01 | **1.24E+01** | 3.75E+01 | 4.08E+01 | 2.17E+01 |
| | Min | 7.95E+02 | 8.02E+02 | 8.69E+02 | 1.30E+03 | 8.20E+02 | 8.44E+02 | 1.38E+03 | 8.82E+02 | **7.56E+02** |
| | Rank | 2 | 4 | 6 | 8 | 3 | 5 | 9 | 7 | **1** |
| $f_8$ | Mean | 9.20E+02 | 9.07E+02 | 9.60E+02 | 1.13E+03 | 9.16E+02 | 9.38E+02 | 1.15E+03 | 9.58E+02 | **8.68E+02** |
| | Std | 1.61E+01 | 3.08E+01 | 2.50E+01 | 1.61E+01 | 1.16E+01 | **1.13E+01** | 2.46E+01 | 1.32E+01 | 2.17E+01 |
| | Min | 8.88E+02 | 8.51E+02 | 9.19E+02 | 1.09E+03 | 8.88E+02 | 9.02E+02 | 1.08E+03 | 9.24E+02 | **8.33E+02** |
| | Rank | 4 | 2 | 7 | 8 | 3 | 5 | 9 | 6 | **1** |
| $f_9$ | Mean | 3.53E+03 | 2.74E+03 | 6.40E+03 | 1.06E+04 | 1.10E+03 | 1.77E+03 | 1.23E+04 | 4.21E+03 | **1.05E+03** |
| | Std | 9.24E+02 | 1.41E+03 | 1.94E+03 | 9.66E+02 | **1.65E+02** | 4.27E+02 | 1.16E+03 | 3.99E+02 | 1.82E+02 |
| | Min | 1.75E+03 | 9.81E+02 | 3.14E+03 | 7.92E+03 | 9.21E+02 | 1.18E+03 | 9.81E+03 | 3.25E+03 | **9.07E+02** |
| | Rank | 5 | 4 | 7 | 8 | 2 | 3 | 9 | 6 | **1** |
| $f_{10}$ | Mean | **3.99E+03** | 4.59E+03 | 5.38E+03 | 8.87E+03 | 5.22E+03 | 5.49E+03 | 8.96E+03 | 5.13E+03 | 4.49E+03 |
| | Std | 3.38E+02 | 6.75E+02 | 2.20E+02 | 3.53E+02 | 2.96E+02 | 3.37E+02 | 4.54E+02 | 4.99E+02 | 8.35E+02 |
| | Min | 3.16E+03 | 2.86E+03 | 4.90E+03 | 7.84E+03 | 4.36E+03 | 4.62E+03 | 7.92E+03 | 4.19E+03 | **2.74E+03** |
| | Rank | **1** | 3 | 6 | 8 | 5 | 7 | 9 | 4 | 2 |
| $f_{11}$ | Mean | 2.69E+03 | 1.28E+03 | 1.25E+03 | 7.87E+03 | 1.22E+03 | 1.45E+03 | 1.07E+04 | 4.35E+03 | **1.22E+03** |
| | Std | 1.12E+03 | 4.93E+01 | 2.94E+01 | 1.68E+03 | **2.39E+01** | 1.63E+02 | 2.21E+03 | 1.01E+03 | 5.02E+01 |
| | Min | 1.28E+03 | 1.18E+03 | 1.19E+03 | 4.70E+03 | 1.15E+03 | 1.25E+03 | 6.37E+03 | 2.55E+03 | **1.14E+03** |
| | Rank | 6 | 4 | 3 | 8 | 2 | 5 | 9 | 7 | **1** |
| $f_{12}$ | Mean | 2.12E+06 | 5.76E+06 | 7.55E+09 | 1.23E+10 | 1.72E+06 | 8.86E+06 | 1.37E+10 | 9.43E+07 | **1.51E+06** |
| | Std | **1.21E+06** | 4.90E+06 | 4.03E+09 | 3.76E+09 | 7.77E+05 | 5.54E+06 | 2.16E+09 | 8.91E+07 | 1.37E+06 |
| | Min | 4.62E+05 | 5.89E+05 | 1.23E+07 | 6.14E+09 | 3.16E+05 | 2.27E+06 | 7.38E+09 | 1.12E+06 | **4.83E+04** |
| | Rank | 3 | 4 | 7 | 8 | 2 | 5 | 9 | 6 | **1** |
| $f_{13}$ | Mean | 1.01E+05 | 8.50E+04 | 2.99E+09 | 1.13E+10 | 4.99E+04 | 1.63E+06 | 5.78E+09 | 3.25E+04 | **1.56E+04** |
| | Std | 8.63E+04 | 5.49E+04 | 4.48E+09 | 5.61E+09 | 9.09E+04 | 1.31E+06 | 1.64E+09 | **7.07E+03** | 1.11E+04 |
| | Min | 1.03E+04 | 1.53E+04 | 3.36E+04 | 2.33E+09 | 4.23E+03 | 5.63E+04 | 3.20E+09 | 1.95E+04 | **1.56E+03** |
| | Rank | 5 | 4 | 7 | 9 | 3 | 6 | 8 | 2 | **1** |
| $f_{14}$ | Mean | 3.70E+05 | 2.42E+04 | **1.52E+03** | 6.91E+06 | 4.27E+03 | 1.86E+05 | 2.06E+07 | 1.07E+06 | 5.00E+03 |
| | Std | 2.14E+05 | 2.14E+04 | **1.75E+01** | 7.87E+06 | 2.34E+03 | 1.51E+05 | 1.43E+07 | 2.60E+05 | 4.87E+03 |
| | Min | 4.03E+04 | 1.79E+03 | **1.49E+03** | 3.51E+05 | 1.67E+03 | 8.81E+03 | 2.77E+06 | 5.99E+05 | 1.58E+03 |
| | Rank | 6 | 4 | **1** | 8 | 2 | 5 | 9 | 7 | 3 |
| $f_{15}$ | Mean | 2.92E+04 | 6.24E+04 | **2.18E+03** | 7.53E+08 | 5.29E+03 | 3.04E+05 | 7.84E+08 | 1.51E+04 | 3.40E+03 |
| | Std | **2.20E+04** | 4.05E+04 | 2.41E+02 | 3.70E+08 | 3.62E+03 | 3.99E+05 | 2.37E+08 | 3.39E+03 | 2.40E+03 |
| | Min | 3.60E+03 | 6.43E+03 | 1.81E+03 | 2.18E+08 | 1.95E+03 | 1.29E+04 | 3.78E+08 | 8.00E+03 | **1.60E+03** |
| | Rank | 5 | 6 | **1** | 8 | 3 | 7 | 9 | 4 | 2 |
| $f_{16}$ | Mean | **2.48E+03** | 2.57E+03 | 2.84E+03 | 7.17E+03 | 2.69E+03 | 2.79E+03 | 6.17E+03 | 3.49E+03 | 2.49E+03 |
| | Std | 1.58E+02 | 3.36E+02 | **1.51E+02** | 1.59E+03 | 1.67E+02 | 1.70E+02 | 9.14E+02 | 3.95E+02 | 2.46E+02 |
| | Min | 2.12E+03 | **1.80E+03** | 2.42E+03 | 4.51E+03 | 2.36E+03 | 2.44E+03 | 4.67E+03 | 2.50E+03 | 1.98E+03 |
| | Rank | **1** | 3 | 6 | 9 | 4 | 5 | 8 | 7 | 2 |
| $f_{17}$ | Mean | 2.08E+03 | 2.03E+03 | 2.12E+03 | 8.57E+03 | **2.02E+03** | 2.15E+03 | 1.38E+04 | 2.89E+03 | 2.05E+03 |
| | Std | 9.50E+01 | 1.73E+02 | 1.06E+02 | 6.19E+03 | 9.63E+01 | 1.26E+02 | 1.83E+04 | 2.13E+02 | 1.76E+02 |
| | Min | 1.88E+03 | 1.79E+03 | 1.90E+03 | 3.47E+03 | 1.84E+03 | 1.84E+03 | 3.16E+03 | 2.37E+03 | **1.77E+03** |
| | Rank | 4 | 2 | 5 | 8 | **1** | 6 | 9 | 7 | 3 |
| $f_{18}$ | Mean | 6.96E+05 | 3.94E+05 | **6.47E+04** | 1.13E+08 | 1.38E+05 | 9.94E+05 | 1.24E+08 | 4.35E+05 | 1.20E+05 |
| | Std | 3.54E+05 | 2.63E+05 | **2.71E+04** | 1.09E+08 | 7.72E+04 | 5.81E+05 | 1.28E+08 | 2.20E+05 | 1.19E+05 |
| | Min | 2.06E+05 | 2.46E+04 | 2.96E+04 | 6.36E+06 | 5.06E+04 | 1.28E+05 | 7.70E+06 | 1.45E+05 | **2.74E+04** |
| | Rank | 6 | 4 | **1** | 8 | 3 | 7 | 9 | 5 | 2 |
| $f_{19}$ | Mean | 5.67E+04 | 1.12E+06 | **2.16E+03** | 7.94E+08 | 9.41E+03 | 3.35E+05 | 5.37E+08 | 1.69E+05 | 6.33E+03 |
| | Std | 4.76E+04 | 7.01E+05 | 3.12E+02 | 4.26E+08 | 7.59E+03 | 3.46E+05 | 2.93E+08 | 8.59E+04 | 4.51E+03 |
| | Min | 5.96E+03 | 5.77E+04 | 1.99E+03 | 1.27E+08 | 2.20E+03 | 1.25E+04 | 1.64E+08 | 3.90E+04 | **1.92E+03** |
| | Rank | 4 | 7 | **1** | 9 | 3 | 6 | 8 | 5 | 2 |
| $f_{20}$ | Mean | 2.38E+03 | 2.43E+03 | 2.59E+03 | 2.96E+03 | 2.38E+03 | 2.45E+03 | 3.19E+03 | 3.07E+03 | **2.30E+03** |
| | Std | **1.11E+02** | 1.47E+02 | 9.16E+01 | 1.34E+02 | **1.11E+02** | 9.30E+01 | 1.83E+02 | 2.43E+02 | 1.37E+02 |
| | Min | 2.11E+03 | 2.12E+03 | 2.35E+03 | 2.64E+03 | 2.13E+03 | 2.22E+03 | 2.74E+03 | 2.52E+03 | **2.06E+03** |
| | Rank | 2 | 4 | 6 | 7 | 3 | 5 | 9 | 8 | **1** |

(*continued on next page*)

**Table 4** (*continued*)

| | | ABC | VSA | CSA | BOA | BSA | DSA | OSA | GSA | MPSO |
|---|---|---|---|---|---|---|---|---|---|---|
| $f_{21}$ | Mean | 2.38E+03 | 2.40E+03 | 2.46E+03 | 2.61E+03 | 2.41E+03 | 2.44E+03 | 2.76E+03 | 2.63E+03 | **2.36E+03** |
| | Std | 7.20E+01 | 2.55E+01 | 3.31E+01 | 1.33E+02 | 1.34E+01 | **1.32E+01** | 4.34E+01 | 3.11E+01 | 1.71E+01 |
| | Min | **2.23E+03** | 2.35E+03 | 2.34E+03 | 2.32E+03 | 2.37E+03 | 2.40E+03 | 2.63E+03 | 2.56E+03 | 2.33E+03 |
| | Rank | 2 | 3 | 6 | 7 | 4 | 5 | 9 | 8 | **1** |
| $f_{22}$ | Mean | 2.70E+03 | **2.45E+03** | 5.47E+03 | 5.37E+03 | 3.06E+03 | 2.83E+03 | 1.01E+04 | 7.42E+03 | 2.79E+03 |
| | Std | 1.02E+03 | 7.80E+02 | 1.98E+03 | 8.75E+02 | 1.66E+03 | 1.08E+03 | 6.32E+02 | **4.49E+02** | 1.26E+03 |
| | Min | 2.31E+03 | **2.30E+03** | 2.33E+03 | 4.08E+03 | 2.31E+03 | 2.34E+03 | 7.51E+03 | 6.34E+03 | **2.30E+03** |
| | Rank | 2 | **1** | 7 | 6 | 5 | 4 | 9 | 8 | 3 |
| $f_{23}$ | Mean | 2.75E+03 | 2.75E+03 | 2.83E+03 | 3.48E+03 | 2.76E+03 | 2.78E+03 | 3.62E+03 | 3.86E+03 | **2.73E+03** |
| | Std | 2.76E+01 | 2.42E+01 | 3.56E+01 | 1.31E+02 | 1.52E+01 | **1.27E+01** | 1.66E+02 | 1.52E+02 | 3.42E+01 |
| | Min | 2.70E+03 | 2.70E+03 | 2.77E+03 | 3.16E+03 | 2.71E+03 | 2.75E+03 | 3.25E+03 | 3.50E+03 | **2.67E+03** |
| | Rank | 3 | 2 | 6 | 7 | 4 | 5 | 8 | 9 | **1** |
| $f_{24}$ | Mean | 2.94E+03 | 2.93E+03 | 2.98E+03 | 4.05E+03 | 2.95E+03 | 2.99E+03 | 3.92E+03 | 3.51E+03 | **2.90E+03** |
| | Std | 1.83E+02 | 3.41E+01 | 4.33E+01 | 2.67E+02 | **1.48E+01** | 1.70E+01 | 1.74E+02 | 8.53E+01 | 3.34E+01 |
| | Min | **2.57E+03** | 2.87E+03 | 2.79E+03 | 3.59E+03 | 2.92E+03 | 2.95E+03 | 3.52E+03 | 3.32E+03 | 2.85E+03 |
| | Rank | 3 | 2 | 5 | 9 | 4 | 6 | 8 | 7 | **1** |
| $f_{25}$ | Mean | 2.89E+03 | 2.90E+03 | **2.89E+03** | 5.67E+03 | 2.90E+03 | 2.90E+03 | 4.76E+03 | 3.00E+03 | 2.95E+03 |
| | Std | 4.90E+00 | 1.40E+01 | **2.04E+00** | 4.79E+02 | 5.67E+00 | 6.49E+00 | 3.00E+02 | 1.76E+01 | 3.33E+01 |
| | Min | 2.89E+03 | 2.89E+03 | **2.88E+03** | 4.78E+03 | 2.89E+03 | 2.89E+03 | 4.28E+03 | 2.96E+03 | 2.89E+03 |
| | Rank | 2 | 3 | **1** | 9 | 4 | 5 | 8 | 7 | 6 |
| $f_{26}$ | Mean | **3.44E+03** | 4.77E+03 | 5.05E+03 | 1.15E+04 | 4.58E+03 | 4.91E+03 | 1.15E+04 | 7.89E+03 | 4.43E+03 |
| | Std | 7.58E+02 | 5.36E+02 | 8.34E+02 | 7.35E+02 | 4.62E+02 | **3.82E+02** | 8.79E+02 | 5.40E+02 | 1.31E+03 |
| | Min | 2.86E+03 | 2.90E+03 | 3.47E+03 | 9.68E+03 | 3.53E+03 | 3.75E+03 | 9.56E+03 | 6.54E+03 | **2.81E+03** |
| | Rank | **1** | 4 | 6 | 9 | 3 | 5 | 8 | 7 | 2 |
| $f_{27}$ | Mean | **3.22E+03** | 3.26E+03 | 3.25E+03 | 4.30E+03 | 3.23E+03 | 3.23E+03 | 4.81E+03 | 5.08E+03 | 3.24E+03 |
| | Std | 6.20E+00 | 2.87E+01 | 1.98E+01 | 4.05E+02 | 5.93E+00 | **4.67E+00** | 4.25E+02 | 3.36E+02 | 1.59E+01 |
| | Min | 3.21E+03 | 3.21E+03 | 3.22E+03 | 3.66E+03 | 3.21E+03 | 3.22E+03 | 3.94E+03 | 4.22E+03 | **3.20E+03** |
| | Rank | **1** | 6 | 5 | 7 | 3 | 2 | 8 | 9 | 4 |
| $f_{28}$ | Mean | 3.25E+03 | 3.23E+03 | **3.23E+03** | 7.85E+03 | 3.29E+03 | 3.30E+03 | 6.75E+03 | 3.58E+03 | 3.35E+03 |
| | Std | 1.52E+01 | 2.11E+01 | 1.50E+01 | 5.42E+02 | 1.38E+01 | **1.16E+01** | 4.80E+02 | 1.86E+02 | 9.10E+01 |
| | Min | 3.22E+03 | 3.19E+03 | **3.20E+03** | 6.57E+03 | 3.24E+03 | 3.28E+03 | 5.85E+03 | 3.33E+03 | 3.26E+03 |
| | Rank | 3 | 2 | **1** | 9 | 4 | 5 | 8 | 7 | 6 |
| $f_{29}$ | Mean | **3.67E+03** | 3.84E+03 | 4.11E+03 | 1.09E+04 | 3.70E+03 | 3.81E+03 | 7.68E+03 | 5.42E+03 | **3.67E+03** |
| | Std | 9.38E+01 | 1.78E+02 | 1.39E+02 | 5.09E+03 | **8.75E+01** | 9.65E+01 | 1.37E+03 | 3.00E+02 | 1.89E+02 |
| | Min | 3.49E+03 | 3.51E+03 | 3.87E+03 | 6.22E+03 | 3.51E+03 | 3.60E+03 | 5.58E+03 | 4.85E+03 | **3.36E+03** |
| | Rank | **1.5** | 5 | 6 | 9 | 3 | 4 | 8 | 7 | **1.5** |
| $f_{30}$ | Mean | 3.73E+04 | 2.93E+06 | 5.70E+04 | 1.80E+09 | 2.98E+04 | 2.67E+05 | 3.60E+09 | 3.54E+06 | **1.21E+04** |
| | Std | 1.58E+04 | 2.19E+06 | 2.39E+04 | 1.01E+09 | 1.62E+04 | 1.86E+05 | 1.04E+09 | 1.29E+06 | **1.20E+04** |
| | Min | 1.34E+04 | 2.68E+05 | 2.15E+04 | 6.07E+08 | 1.04E+04 | 5.42E+04 | 1.77E+09 | 1.31E+06 | **5.60E+03** |
| | Rank | 3 | 6 | 4 | 8 | 2 | 5 | 9 | 7 | **1** |
| Final Rank | | 89.5 | 103 | 139 | 231 | 93 | 148 | 247 | 189 | 65.5 |
| Total Rank | | 2 | 4 | 5 | 8 | 3 | 6 | 9 | 7 | 1 |

**Table 5**
Statistical comparisons of WSRT for MPSO vs. PSO, FDR_PSO, PSO_cf_local and PSO_cf.

| fun | MPSO vs. PSO | | | | MPSO vs. FDR_PSO | | | | MPSO vs. PSO_cf_local | | | | MPSO vs. PSO_cf | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *p*_Value | R+ | R− | Winner | *p*_Value | R+ | R− | Winner | *p*_Value | R+ | R− | Winner | *p*_Value | R+ | R− | Winner |
| $f_1$ | 7.56E−10 | 0 | 1275 | + | 0.000621 | 992 | 283 | + | 1.86E−05 | 1081 | 194 | − | 9.07E−10 | 3 | 1272 | + |
| $f_3$ | 7.56E−10 | 0 | 1275 | + | 0.000177 | 249 | 1026 | − | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + |
| $f_4$ | 7.56E−10 | 0 | 1275 | + | 0.308478 | 743 | 532 | = | 0.746404 | 671 | 604 | = | 1.15E−09 | 7 | 1268 | + |
| $f_5$ | 7.56E−10 | 0 | 1275 | + | 0.029494 | 863 | 412 | − | 0.336803 | 737 | 538 | = | 1.09E−09 | 6 | 1269 | + |
| $f_6$ | 1.47E−09 | 11 | 1264 | + | 1.09E−09 | 1269 | 6 | − | 7.56E−10 | 1275 | 0 | − | 1.68E−08 | 53 | 1222 | + |
| $f_7$ | 3.57E−09 | 26 | 1249 | + | 0.011592 | 899 | 376 | − | 0.000389 | 1005 | 270 | − | 1.2E−08 | 47 | 1228 | + |
| $f_8$ | 9.63E−10 | 4 | 1271 | + | 0.047284 | 843 | 432 | − | 0.660499 | 592 | 683 | = | 1.66E−09 | 13 | 1262 | + |
| $f_9$ | 7.56E−10 | 0 | 1275 | + | 0.00014 | 1032 | 243 | − | 6.56E−05 | 1051 | 224 | − | 8.03E−10 | 1 | 1274 | + |
| $f_{10}$ | 1.43E−05 | 188 | 1087 | + | 0.926932 | 647 | 628 | = | 0.710153 | 676 | 599 | = | 0.420215 | 554 | 721 | = |
| $f_{11}$ | 7.56E−10 | 0 | 1275 | + | 0.236997 | 515 | 760 | + | 0.0325 | 416 | 859 | + | 2.66E−09 | 21 | 1254 | + |
| $f_{12}$ | 7.56E−10 | 0 | 1275 | + | 0.019739 | 879 | 396 | − | 0.0156 | 387 | 888 | + | 1.02E−09 | 5 | 1270 | + |
| $f_{13}$ | 7.56E−10 | 0 | 1275 | + | 0.000913 | 294 | 981 | + | 2.19E−06 | 147 | 1128 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{14}$ | 7.56E−10 | 0 | 1275 | + | 4.54E−08 | 71 | 1204 | + | 4.25E−09 | 29 | 1246 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{15}$ | 8.03E−10 | 1 | 1274 | + | 4.92E−05 | 217 | 1058 | + | 2.93E−08 | 63 | 1212 | + | 1.56E−09 | 12 | 1263 | + |
| $f_{16}$ | 1.55E−07 | 94 | 1181 | + | 0.448581 | 716 | 559 | = | 0.079762 | 819 | 456 | = | 2.23E−07 | 101 | 1174 | + |
| $f_{17}$ | 2.10E−08 | 57 | 1218 | + | 0.073344 | 823 | 452 | = | 0.055343 | 836 | 439 | = | 1.32E−07 | 91 | 1184 | + |
| $f_{18}$ | 3.78E−09 | 27 | 1248 | + | 1.99E−06 | 145 | 1130 | + | 1.72E−07 | 96 | 1179 | + | 5.87E−07 | 120 | 1155 | + |
| $f_{19}$ | 8.03E−10 | 1 | 1274 | + | 0.269029 | 523 | 752 | = | 0.003089 | 331 | 944 | + | 1.86E−09 | 15 | 1260 | + |
| $f_{20}$ | 1.85E−03 | 315 | 960 | + | 0.996149 | 638 | 637 | = | 0.331969 | 738 | 537 | = | 7.52E−07 | 125 | 1150 | + |
| $f_{21}$ | 7.56E−10 | 0 | 1275 | + | 0.184401 | 500 | 775 | = | 0.000389 | 270 | 1005 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{22}$ | 2.36E−09 | 19 | 1256 | + | 0.00024 | 257 | 1018 | + | 0.005515 | 350 | 925 | + | 2.12E−07 | 100 | 1175 | + |
| $f_{23}$ | 7.56E−10 | 0 | 1275 | + | 1.19E−07 | 89 | 1186 | + | 2.12E−07 | 100 | 1175 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{24}$ | 7.56E−10 | 0 | 1275 | + | 1.09E−09 | 6 | 1269 | + | 6.81E−07 | 123 | 1152 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{25}$ | 4.01E−09 | 28 | 1247 | + | 1.15E−09 | 1268 | 7 | − | 9.63E−10 | 1271 | 4 | + | 0.002811 | 328 | 947 | + |
| $f_{26}$ | 2.66E−09 | 21 | 1254 | + | 0.783225 | 666 | 609 | = | 0.041184 | 426 | 849 | + | 3.85E−08 | 68 | 1207 | + |
| $f_{27}$ | 2.23E−09 | 18 | 1257 | + | 0.034099 | 418 | 857 | + | 0.502282 | 707 | 568 | = | 2.36E−09 | 19 | 1256 | + |
| $f_{28}$ | 7.56E−10 | 0 | 1275 | + | 1.68E−08 | 1222 | 53 | − | 0.681612 | 680 | 595 | = | 1.98E−08 | 56 | 1219 | + |
| $f_{29}$ | 1.13E−08 | 46 | 1229 | + | 0.490062 | 709 | 566 | = | 0.002723 | 327 | 948 | + | 9.07E−10 | 3 | 1272 | + |
| $f_{30}$ | 7.56E−10 | 0 | 1275 | + | 0.002723 | 327 | 948 | + | 2.34E−08 | 59 | 1216 | + | 1.15E−09 | 7 | 1268 | + |
| +/−/= | | | | 29/0/0 | | | | 11/9/9 | | | | 15/4/10 | | | | 28/0/1 |

**Table 6**
Statistical comparisons of WSRT for MPSO vs. FIPS, CLPSO, SPSO2011 and DNLPSO.

| fun | MPSO vs. FIPS | | | | MPSO vs. CLPSO | | | | MPSO vs. SPSO2011 | | | | MPSO vs. DNLPSO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $p$_Value | R+ | R− | Winner | $p$_Value | R+ | R− | Winner | $p$_Value | R+ | R− | Winner | $p$_Value | R+ | R− | Winner |
| $f_1$ | 7.41E−05 | 1048 | 227 | − | 0.101791 | 468 | 807 | = | 9.63E−10 | 1271 | 4 | − | 8.03E−10 | 1 | 1274 | + |
| $f_3$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 2.66E−09 | 21 | 1254 | + | 1.66E−09 | 13 | 1262 | + |
| $f_4$ | 0.099788 | 808 | 467 | = | 1.15E−09 | 7 | 1268 | + | 0.001976 | 958 | 317 | − | 8.03E−10 | 1 | 1274 | + |
| $f_5$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.23E−09 | 8 | 1267 | + | 3.45E−08 | 66 | 1209 | + |
| $f_6$ | 1.15E−09 | 1268 | 7 | − | 0.0274 | 866 | 409 | − | 0.114494 | 801 | 474 | = | 4.86E−06 | 164 | 1111 | + |
| $f_7$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.76E−09 | 14 | 1261 | + | 2.53E−06 | 150 | 1125 | + |
| $f_8$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 8.53E−10 | 2 | 1273 | + | 2.12E−07 | 100 | 1175 | + |
| $f_9$ | 3.78E−09 | 1248 | 27 | − | 7.56E−10 | 0 | 1275 | + | 1.63E−05 | 1084 | 191 | − | 7.56E−10 | 0 | 1275 | + |
| 10 | 7.56E−10 | 0 | 1275 | + | 1.38E−09 | $f_{10}$ | 1265 | + | 7.56E−10 | 0 | 1275 | + | 5.05E−07 | 117 | 1158 | + |
| $f_{11}$ | 3.51E−06 | 157 | 1118 | + | 7.56E−10 | 0 | 1275 | + | 0.135848 | 483 | 792 | = | 1.15E−09 | 7 | 1268 | + |
| $f_{12}$ | 1.78E−05 | 193 | 1082 | + | 7.56E−10 | 0 | 1275 | + | 0.059131 | 833 | 442 | = | 1.23E−09 | 8 | 1267 | + |
| $f_{13}$ | 1.02E−07 | 86 | 1189 | + | 7.56E−10 | 0 | 1275 | + | 7.38E−08 | 80 | 1195 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{14}$ | 7.56E−10 | 0 | 1275 | + | 8.03E−10 | 1 | 1274 | + | 4.51E−09 | 30 | 1245 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{15}$ | 3.17E−09 | 24 | 1251 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{16}$ | 5.95E−08 | 76 | 1199 | + | 0.72458 | 674 | 601 | = | 0.003722 | 337 | 938 | + | 1.17E−06 | 134 | 1141 | + |
| $f_{17}$ | 0.000579 | 994 | 281 | + | 0.0006 | 993 | 282 | − | 0.065923 | 828 | 447 | = | 1.07E−08 | 45 | 1230 | + |
| $f_{18}$ | 1.47E−09 | 11 | 1264 | + | 7.58E−09 | 39 | 1236 | + | 5.58E−07 | 119 | 1156 | + | 1.38E−09 | 10 | 1265 | + |
| $f_{19}$ | 0.026076 | 407 | 868 | + | 1.66E−09 | 13 | 1262 | + | 7.56E−10 | 0 | 1275 | + | 1.86E−09 | 15 | 1260 | + |
| $f_{20}$ | 0.000115 | 238 | 1037 | + | 0.646571 | 590 | 685 | = | 0.000102 | 235 | 1040 | + | 4.95E−02 | 434 | 841 | + |
| $f_{21}$ | 7.56E−10 | 0 | 1275 | + | 5.06E−09 | 32 | 1243 | = | 9.63E−10 | 4 | 1271 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{22}$ | 0.0003 | 1012 | 263 | − | 0.000434 | 273 | 1002 | + | 9.63E−10 | 1271 | 4 | − | 3.37E−09 | 25 | 1250 | + |
| $f_{23}$ | 8.53E−10 | 2 | 1273 | + | 1.76E−09 | 14 | 1261 | + | 0.000375 | 269 | 1006 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{24}$ | 8.03E−10 | 1 | 1274 | + | 1.15E−09 | 7 | 1268 | + | 0.207759 | 507 | 768 | = | 7.56E−10 | 0 | 1275 | + |
| $f_{25}$ | 8.03E−10 | 1274 | 1 | − | 3.2E−06 | 1120 | 155 | − | 2.53E−06 | 1125 | 150 | − | 9.06E−05 | 232 | 1043 | + |
| $f_{26}$ | 2.48E−08 | 1215 | 60 | − | 0.070297 | 450 | 825 | = | 0.72458 | 601 | 674 | = | 2.22E−08 | 58 | 1217 | + |
| $f_{27}$ | 1.66E−09 | 1262 | 13 | − | 0.533523 | 702 | 573 | = | 0.121297 | 477 | 798 | = | 1.86E−09 | 15 | 1260 | + |
| $f_{28}$ | 7.56E−10 | 1275 | 0 | − | 0.702977 | 677 | 598 | = | 7.56E−10 | 1275 | 0 | − | 8.53E−10 | 2 | 1273 | + |
| $f_{29}$ | 8.5E−09 | 41 | 1234 | + | 0.016447 | 389 | 886 | + | 0.000979 | 296 | 979 | + | 1.81E−07 | 97 | 1178 | + |
| $f_{30}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + |
| +/−/= | | | | 20/8/1 | | | | 19/3/7 | | | | 17/6/6 | | | | 29/0/0 |

**Table 7**
Statistical comparisons of WSRT for MPSO vs. ABC, BOA, ASO and BSA.

| fun | MPSO vs. ABC | | | | MPSO vs. BOA | | | | MPSO vs. CSA | | | | MPSO vs. BSA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $p$_Value | R+ | R− | Winner | $p$_Value | R+ | R− | Winner | $p$_Value | R+ | R− | Winner | $p$_Value | R+ | R− | Winner |
| $f_1$ | 8.03E−10 | 1274 | 1 | − | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | − | 4.01E−09 | 1247 | 28 | − |
| $f_3$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | − | 7.56E−10 | 0 | 1275 | + |
| $f_4$ | 4.3E−08 | 1205 | 70 | − | 7.56E−10 | 0 | 1275 | + | 4.50E−04 | 1001 | 274 | − | 0.000361 | 268 | 1007 | + |
| $f_5$ | 5.37E−09 | 33 | 1242 | = | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 8.03E−10 | 1 | 1274 | + |
| $f_6$ | 7.56E−10 | 1275 | 0 | − | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.3E−09 | 1266 | 9 | − |
| $f_7$ | 2.78E−06 | 152 | 1123 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 8.53E−10 | 2 | 1273 | + |
| $f_8$ | 9.63E−10 | 4 | 1271 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.15E−09 | 7 | 1268 | + |
| $f_9$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 0.039305 | 424 | 851 | + |
| $f_{10}$ | 0.001285 | 971 | 304 | − | 7.56E−10 | 0 | 1275 | + | 1.40E−07 | 92 | 1183 | + | 1.2E−05 | 184 | 1091 | + |
| $f_{11}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.20E−04 | 239 | 1036 | + | 0.313086 | 533 | 742 | − |
| $f_{12}$ | 0.009545 | 369 | 906 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 0.140987 | 485 | 790 | = |
| $f_{13}$ | 3.57E−09 | 26 | 1249 | + | 7.56E−10 | 0 | 1275 | + | 8.03E−10 | 1 | 1274 | + | 0.000289 | 262 | 1013 | + |
| $f_{14}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 1275 | 0 | + | 0.681612 | 595 | 680 | = |
| $f_{15}$ | 1.66E−09 | 13 | 1262 | + | 7.56E−10 | 0 | 1275 | + | 1.12E−03 | 975 | 300 | − | 0.001122 | 300 | 975 | + |
| $f_{16}$ | 0.973047 | 634 | 641 | = | 7.56E−10 | 0 | 1275 | + | 5.69E−09 | 34 | 1241 | + | 0.00012 | 239 | 1036 | + |
| $f_{17}$ | 0.107995 | 471 | 804 | = | 7.56E−10 | 0 | 1275 | + | 6.44E−04 | 284 | 991 | + | 0.377088 | 729 | 546 | = |
| $f_{18}$ | 1.09E−09 | 6 | 1269 | + | 7.56E−10 | 0 | 1275 | + | 6.83E−05 | 1050 | 225 | − | 0.045171 | 430 | 845 | + |
| $f_{19}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.02E−09 | 1270 | 5 | − | 0.021318 | 399 | 876 | + |
| $f_{20}$ | 0.003089 | 331 | 944 | = | 7.56E−10 | 0 | 1275 | + | 2.99E−09 | 23 | 1252 | + | 0.004471 | 343 | 932 | + |
| $f_{21}$ | 0.083138 | 458 | 817 | + | 1.02E−09 | 5 | 1270 | + | 7.56E−10 | 0 | 1275 | + | 8.03E−10 | 1 | 1274 | + |
| $f_{22}$ | 0.233191 | 514 | 761 | = | 5.37E−09 | 33 | 1242 | + | 1.64E−06 | 141 | 1134 | + | 0.071808 | 451 | 824 | + |
| $f_{23}$ | 0.0006 | 282 | 993 | + | 7.56E−10 | 0 | 1275 | + | 1.15E−09 | 7 | 1268 | + | 3.05E−06 | 154 | 1121 | + |
| $f_{24}$ | 0.313086 | 533 | 742 | = | 7.56E−10 | 0 | 1275 | + | 1.42E−08 | 50 | 1225 | + | 1.98E−08 | 56 | 1219 | + |
| $f_{25}$ | 8.53E−10 | 1273 | 2 | − | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 1275 | 0 | − | 1.09E−09 | 1269 | 6 | − |
| $f_{26}$ | 0.00024 | 1018 | 257 | − | 7.56E−10 | 0 | 1275 | + | 7.03E−02 | 450 | 825 | = | 0.382324 | 547 | 728 | = |
| $f_{27}$ | 3.45E−08 | 1209 | 66 | − | 7.56E−10 | 0 | 1275 | + | 8.49E−02 | 459 | 816 | = | 0.000198 | 1023 | 252 | − |
| $f_{28}$ | 8.03E−10 | 1274 | 1 | − | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 1275 | 0 | − | 3.09E−08 | 1211 | 64 | − |
| $f_{29}$ | 0.919265 | 648 | 627 | = | 7.56E−10 | 0 | 1275 | + | 1.30E−09 | 9 | 1266 | + | 0.502282 | 568 | 707 | = |
| $f_{30}$ | 1.27E−08 | 48 | 1227 | + | 7.56E−10 | 0 | 1275 | + | 1.30E−09 | 9 | 1266 | + | 8.66E−08 | 83 | 1192 | + |
| +/−/= | | | | 14/8/7 | | | | 29/0/0 | | | | 19/8/2 | | | | 18/6/5 |

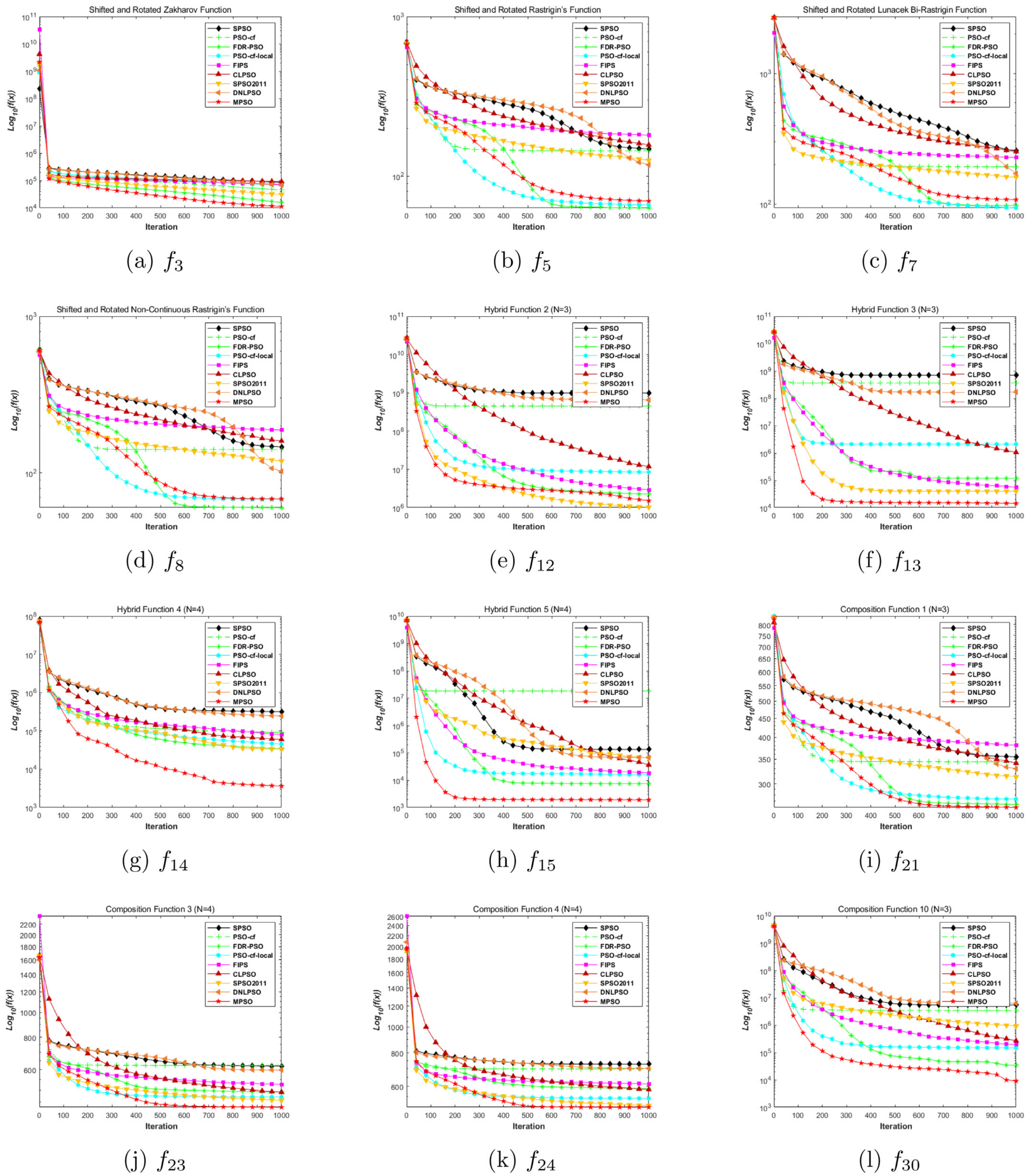
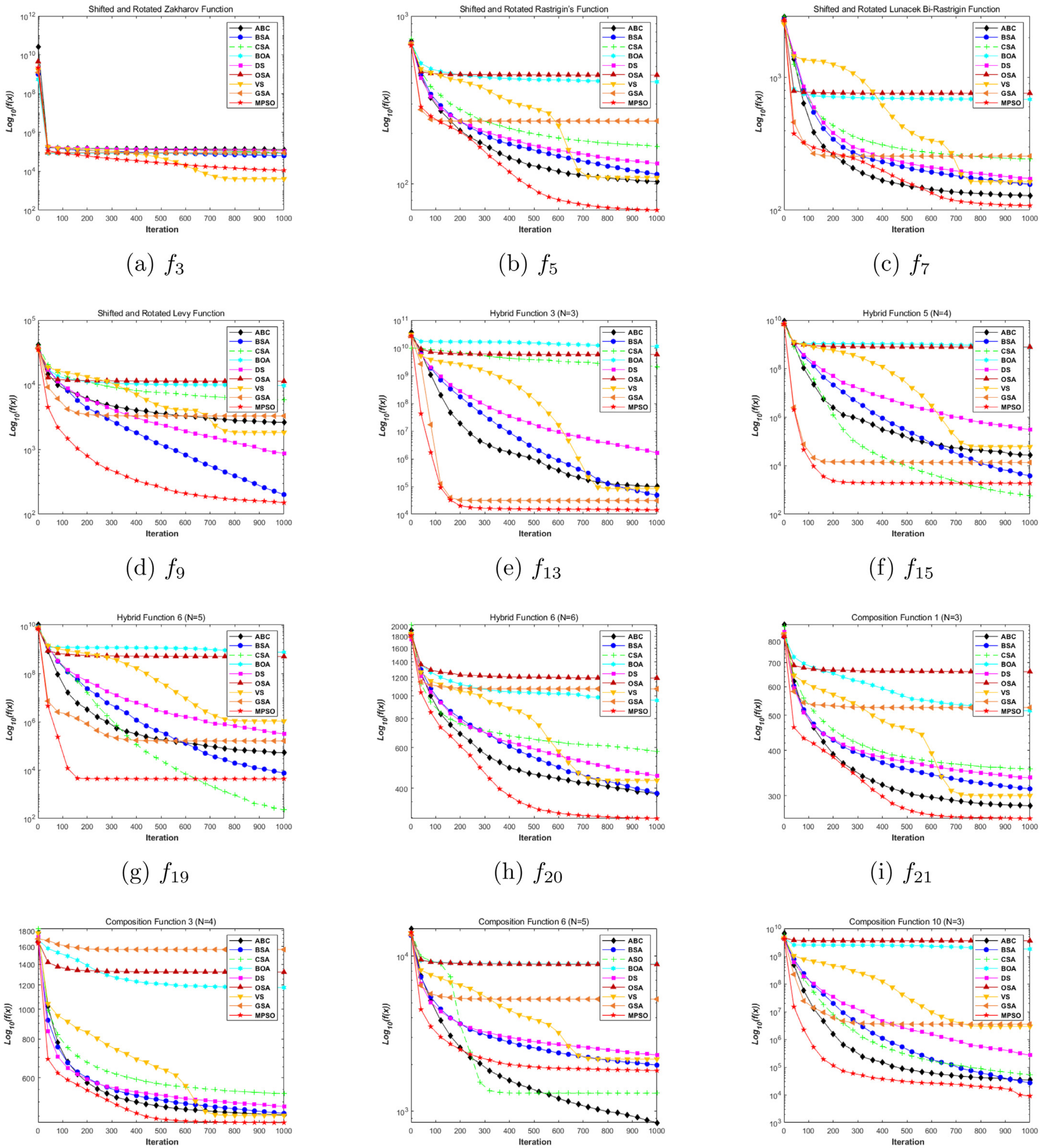



**Fig. 6.** Convergence curve of algorithms for variants of PSO.

Both Friedman-test and Quad-test are non-parametric statistical test (Derrac, García, Molina, & Herrera, 2011). Generally, they are used to detect differences in treatments across multiple test attempts. In this study, they are adopted to compare the comprehensive performance of each algorithm on a set of functions. Thus, the result of the Friedman-test and Quad-test provide an overview of algorithm's performance. Tables 9 and 10 depict the ranks calculated through them for all benchmark functions among all algorithms. As shown in Tables 9 and 10, MPSO ranks first among the 17 algorithms. The $p-value$ computed from the statistics of the Friedman-test and Quade-test strongly indicates that there are significant differences among the 17 algorithms.




(a) $f_3$ (b) $f_5$ (c) $f_7$

(d) $f_9$ (e) $f_{13}$ (f) $f_{15}$

(g) $f_{19}$ (h) $f_{20}$ (i) $f_{21}$

(j) $f_{23}$ (k) $f_{26}$ (l) $f_{30}$

**Fig. 7.** Convergence curve of algorithms for 8 evolutionary algorithms.

**Table 8**
Statistical comparisons of WSRT for MPSO vs. DSA, GSA, OSA and VSA.

| fun | MPSO vs. DSA | | | | MPSO vs. GSA | | | | MPSO vs. OSA | | | | MPSO vs. VSA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | p_Value | R+ | R− | Winner | p_Value | R+ | R− | Winner | p_Value | R+ | R− | Winner | p_Value | R+ | R− | Winner |
| $f_1$ | 9.53E−09 | 1232 | 43 | − | 0.090237 | 462 | 813 | = | 7.56E−10 | 0 | 1275 | + | 8.53E−10 | 1273 | 2 | − |
| $f_3$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.27E−08 | 1227 | 48 | + |
| $f_4$ | 2.41E−05 | 200 | 1075 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 0.790651 | 665 | 610 | = |
| $f_5$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.13E−08 | 46 | 1229 | + |
| $f_6$ | 7.56E−10 | 1275 | 0 | − | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.76E−09 | 14 | 1261 | + |
| $f_7$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.27E−08 | 48 | 1227 | + |
| $f_8$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.63E−07 | 95 | 1180 | + |
| $f_9$ | 9.63E−10 | 4 | 1271 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 8.03E−10 | 1 | 1274 | + |
| $f_{10}$ | 3.45E−08 | 66 | 1209 | + | 1.49E−05 | 189 | 1086 | + | 7.56E−10 | 0 | 1275 | + | 0.387605 | 548 | 727 | = |
| $f_{11}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 3.92E−07 | 112 | 1163 | + |
| $f_{12}$ | 1.09E−09 | 6 | 1269 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 2.12E−07 | 100 | 1175 | + |
| $f_{13}$ | 7.56E−10 | 0 | 1275 | + | 3.45E−08 | 66 | 1209 | + | 7.56E−10 | 0 | 1275 | + | 9.07E−10 | 3 | 1272 | + |
| $f_{14}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 2.23E−07 | 101 | 1174 | + |
| $f_{15}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{16}$ | 1.55E−07 | 94 | 1181 | + | 8.03E−10 | 1 | 1274 | + | 7.56E−10 | 0 | 1275 | + | 0.294924 | 529 | 746 | = |
| $f_{17}$ | 0.005196 | 348 | 927 | + | 8.03E−10 | 1 | 1274 | + | 7.56E−10 | 0 | 1275 | + | 0.843129 | 658 | 617 | = |
| $f_{18}$ | 9.63E−10 | 4 | 1271 | + | 3.09E−08 | 64 | 1211 | + | 7.56E−10 | 0 | 1275 | + | 1.19E−07 | 89 | 1186 | + |
| $f_{19}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + |
| $f_{20}$ | 7.52E−07 | 125 | 1150 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 3.53E−05 | 209 | 1066 | + |
| $f_{21}$ | 8.03E−10 | 1 | 1274 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.07E−08 | 45 | 1230 | + |
| $f_{22}$ | 0.000466 | 275 | 1000 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 1.63E−07 | 1180 | 95 | − |
| $f_{23}$ | 4.51E−09 | 30 | 1245 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 0.00012 | 239 | 1036 | = |
| $f_{24}$ | 1.02E−09 | 5 | 1270 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 0.000621 | 283 | 992 | + |
| $f_{25}$ | 1.15E−09 | 1268 | 7 | − | 1.42E−08 | 50 | 1225 | + | 7.56E−10 | 0 | 1275 | + | 8.53E−10 | 1273 | 2 | − |
| $f_{26}$ | 0.017335 | 391 | 884 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 0.078116 | 455 | 820 | = |
| $f_{27}$ | 1E-05 | 1095 | 180 | − | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 0.001851 | 315 | 960 | + |
| $f_{28}$ | 2.09E−06 | 1129 | 146 | − | 8.03E−10 | 1 | 1274 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 1275 | 0 | − |
| $f_{29}$ | 6.3E−05 | 223 | 1052 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 2.41E−05 | 200 | 1075 | + |
| $f_{30}$ | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + | 7.56E−10 | 0 | 1275 | + |
| +/−/= | | | | 24/5/0 | | | | 28/0/1 | | | | 29/0/0 | | | | 19/4/6 |

**Table 9**
Average Rankings of the algorithms (Friedman-test).

| Algorithm | Ranking | Final rank |
|---|---|---|
| OSA | 1.4655172413793098 | 17 |
| BOA | 2.189655172413793 | 16 |
| SPSO | 4.862068965517242 | 15 |
| GSA | 5.379310344827586 | 14 |
| PSO_cf | 6.344827586206897 | 13 |
| DNLPSO | 6.517241379310345 | 12 |
| CSA | 8.689655172413795 | 11 |
| DSA | 8.89655172413793 | 10 |
| CLPSO | 9.499999999999998 | 9 |
| FIPS | 10.603448275862068 | 8 |
| VSA | 11.413793103448274 | 7 |
| SPSO2011 | 12.086206896551722 | 6 |
| ABC | 12.29310344827586 | 5 |
| PSO_cf_local | 12.586206896551722 | 4 |
| BSA | 12.79310344827586 | 3 |
| FDR_PSO | 13.448275862068966 | 2 |
| **MPSO** | **13.931034482758621** | **1** |
| *ρ_value* | 1.3818635125062428E−10 | |

**Table 10**
Average Rankings of the algorithms (Quade-test).

| Algorithm | Ranking | Final rank |
|---|---|---|
| OSA | 1.4919540229885055 | 17 |
| BOA | 2.1770114942528735 | 16 |
| SPSO | 4.4689655172413785 | 15 |
| DNLPSO | 5.610344827586207 | 14 |
| PSO_cf | 6.1873563218390775 | 13 |
| GSA | 6.59080459770115 | 12 |
| DS | 8.417241379310346 | 11 |
| CSA | 9.427586206896551 | 10 |
| CLPSO | 9.697701149425288 | 9 |
| VS | 11.418390804597701 | 8 |
| PSO_cf_local | 11.525287356321837 | 7 |
| FIPS | 11.566666666666668 | 6 |
| ABC | 11.639080459770113 | 5 |
| SPSO2011 | 12.372413793103448 | 4 |
| FDR_PSO | 13.022988505747128 | 3 |
| BSA | 13.259770114942533 | 2 |
| **MPSO** | **14.126436781609195** | **1** |
| *ρ_value* | 1.179738817334863E−50 | |

## 5. Conclusions

In this paper, to overcome the drawbacks of PSO such as premature convergence, poor balance between global exploration and local exploitation, and poor ability in solving complex optimization problems in expert systems, a modified PSO called MPSO is proposed. In MPSO, stochastic and mainstream learning strategies are devised to avoid premature convergence.Meanwhile, a chaos-based non-linear inertia weight is proposed to balance exploration and exploitation better by enlarging or shrinking the search step. In additions, an adaptive position updating strategy and terminal replacement mechanism is adopted to enhance the ability of PSO to solve complex optimization problems in expert systems. To verify MPSO's convergence performance on various optimization problems, comparison experiments on complex CEC2017 are carried out. The experiment results and corresponding statistical analysis indicate that MPSO outperforms quite a few state-of-the-art PSO variants and some other evolutionary algorithms proposed recently.

However, MPSO still has some limitations.

(1) Although MPSO has achieved excellent performance, its convergence and stability are lack of theoretical support.

(2) There are some parameters in MPSO, which limit the applicability of MPSO, because these parameters need to be adjusted for different real optimization problems in expert systems. This increases the difficulty of problem solving.
(3) According to the experimental results, it can be found that the performance of MPSO is ideal, but there are still some defects. For example, the performance of MPSO on functions $f_1$, $f_{17}$ and $f_{25}$ are not ideal, which indicates the deficiency of MPSO performance. In the future, we will work to further improve the performance of MPSO.

The future work will be carried out from the following aspects.

(1) Using Markov chain theory to analyze the convergence of MPSO, and using fixed point theorem to analyze the first and second order stability of MPSO. However, the second-order stability of PSO without stagnation assumption is complex and difficult. It may be possible to analyze the second-order stability on the premise of stagnation assumption, which should be solved in the future.
(2) As we all know, the parameters of the algorithm will increase the difficulty of solving practical optimization problems in expert systems. In the future, MPSO will be further improved and its parameters will be reduced, so as to reduce its difficulty in solving practical problems.
(3) MPSO will be combined with neural network, that is, using MPSO algorithm to train artificial neural network, which is widely used in expert system.
(4) Because the optimization problem in expert system often has constraints, and to deal with the constrained optimization problem, we need to maintain the feasible solution in the algorithm. Therefore, in the future, a feasible solution maintenance method will be designed and used in MPSO to solve constraint optimization problems

## Declaration of Competing Interest

No conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication.

## Credit authorship contribution statement

**Hao Liu:** Conceptualization, Methodology, Software, Resources, Writing - original draft, Writing - review & editing, Formal analysis, Supervision, Project administration, Funding acquisition. **Xu-Wei Zhang:** Methodology, Investigation, Data curation, Software, Visualization, Writing - review & editing, Conceptualization. **Liang-Ping Tu:** Formal analysis, Resources, Project administration, Funding acquisition.


## Acknowledgements

The authors wish to acknowledge the National Natural Science Foundation of China (Grant no. U1731128); the Natural Science Foundation of Liaoning Province (Grant no. 2019-MS-174); the Foundation of Liaoning Province Education Administration (Grant no. 2019LNJC12) for the financial support; the Graduate Science and Technology Innovation Program of USTL (Grant no. LKDYC201921).


## Appendix A

Shifted and Rotated Bent Cigar Function ($F_1$)

$$F_1(x) = f_1(M(x - o_1)) + F_1^*$$

Shifted and Rotated Sum of Different Power Function ($F_2$)

$$F_2(x) = f_2(M(x - o_2)) + F_2^*$$

Shifted and Rotated Zakharov Function ($F_3$)

$$F_3(x) = f_3(M(x - o_3)) + F_3^*$$

Shifted and Rotated Rosenbrock's Function($F_4$)

$$F_4(x) = f_4(M(x - o_4)) + F_4^*$$

Shifted and Rotated Rastrigin's Function($F_5$)

$$F_5(x) = f_5(M(x - o_5)) + F_5^*$$

Shifted and Rotated Schaffer's F7 Function($F_6$)

$$F_6(x) = f_{20}\left(M\frac{0.5(x - o_6)}{100}\right) + F_6^*$$

Shifted and Rotated Lunacek Bi-Rastrigin's Function($F_7$)

$$F_7(x) = f_7\left(M\frac{600(x - o_7)}{100}\right) + F_7^*$$

Shifted and Rotated Non-Continuous Rastrigin's Function($F_8$)

$$F_8(x) = f_8\left(\frac{5.12(x - o_8)}{100}\right) + F_8^*$$

Shifted and Rotated Levy Function($F_9$)

$$F_9(x) = f_9\left(M\frac{5.12(x - o_9)}{100}\right) + F_9^*$$

Shifted and Rotated Schwefel's Function($F_{10}$)

$$F_{10}(x) = f_{10}\left(\frac{1000(x - o_{10})}{100}\right) + F_{10}^*$$

Hybrid Function 1($F_{11}$)

$$F_{11}(x) = f_3(M_1z_1) + f_4(M_2z_2) + f_5(M_3z_3) + F_{11}^*$$

Hybrid Function 2($F_{12}$)

$$F_{12}(x) = f_{11}(M_1z_1) + f_{10}(M_2z_2) + f_1(M_3z_3) + F_{12}^*$$

Hybrid Function 3($F_{13}$)

$$F_{13}(x) = f_1(M_1z_1) + f_4(M_2z_2) + f_7(M_3z_3) + F_{13}^*$$

Hybrid Function 4 ($F_{14}$)

$$F_{14}(x) = f_{11}(M_1z_1) + f_{13}(M_2z_2) + f_{20}(M_3z_3) + f_5(M_4z_4) + F_{14}^*$$

Hybrid Function 5 ($F_{15}$)

$$F_{15}(x) = f_1(M_1z_1) + f_{18}(M_2z_2) + f_5(M_3z_3) + f_4(M_4z_4) + F_{15}^*$$

Hybrid Function 6 ($F_{16}$)

$$F_{16}(x) = f_6(M_1z_1) + f_{18}(M_2z_2) + f_4(M_3z_3) + f_{10}(M_4z_4) + F_{16}^*$$

Hybrid Function 7 ($F_{17}$)

$$F_{17}(x) = f_{16}(M_1z_1) + f_{13}(M_2z_2) + f_{19}(M_3z_3) + f_{10}(M_4z_4) + f_5(M_5z_5) + F_{17}^*$$

Hybrid Function 8 ($F_{18}$)

$$F_{18}(x) = f_1(M_1z_1) + f_4 13(M_2z_2) + f_5(M_3z_3) + f_{18}(M_4z_4) + f_{12}(M_5z_5) + F_{18}^*$$

Hybrid Function 9 ($F_{19}$)

$$F_{19}(x) = f_1(M_1z_1) + f_5(M_2z_2) + f_{19}(M_3z_3) + f_{14}(M_4z_4) + f_6(M_5z_5) + F_{19}^*$$

Hybrid Function 10 ($F_{20}$)

$$F_{20}(x) = f_{17}(M_1z_1) + f_4 16M_2z_2) + f_{13}(M_3z_3) + f_5(M_4z_4) + f_{10}(M_5z_5) + f_{20}(M_4z_4) + f_5(M_6z_6) + F_{20}^*$$

Composition Function 1 ($F_{21}$)

$$F_{21}(x) = \sum_{i=1}^{3} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{21}^*$$

$$\omega_i = \frac{1}{\sqrt{\sum_{i=1}^{D}(x_j - o_{ij})^2}} exp\left(-\frac{\sum_{i=1}^{D}(x_j - o_{ij})^2}{2D\sigma_i^2}\right)$$

$\sigma = [10, 20, 30], \lambda = [1, 1e-6, 1], bias = [0, 100, 200]$
$g = [f_4', f_{11}, f_4]$

Composition Function 2($F_{22}$)

$$F_{22}(x) = \sum_{i=1}^{3} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{22}^*$$

$\sigma = [10, 20, 30], \lambda = [1, 10, 1], bias = [0, 100, 200],$
$g = [f_5', f_{15}', f_{10}']$

Composition Function 3 ($F_{23}$)

$$F_{23}(x) = \sum_{i=1}^{4} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{23}^*$$

$\sigma = [10, 20, 30, 40], \lambda = [1, 10, 1, 1], bias = [0, 100, 200, 300],$
$g = [f_4', f_{13}', f_{10}', f_5']$

Composition Function 4($F_{24}$)

$$F_{24}(x) = \sum_{i=1}^{4} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{24}^*$$

$\sigma = [10, 20, 30, 40], \lambda = [1, 1e-6, 10, 1],$
$bias = [0, 100, 200, 300], g = [f_{13}', f_{11}', F_{15}', f_5']$

Composition Function 5($F_{25}$)

$$F_{25}(x) = \sum_{i=1}^{5} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{25}^*$$

$\sigma = [10, 20, 30, 40, 50], \lambda = [1, 1, 10, 1e-6, 1],$
$bias = [0, 100, 200, 300, 400],$
$g = [f_5', f_{15}', f_5', f_5', f_{10}']$

Composition Function 6($F_{26}$)

$$F_{26}(x) = \sum_{i=1}^{5} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{26}^*$$

$\sigma = [10, 20, 20, 30, 40], \lambda = [1e-26, 10, 1e-6, 10, 5e-4],$
$bias = [0, 100, 200, 300, 400], g = [f_6', f_{10}', f_{15}', f_4', f_5']$

Composition Function 7($F_{27}$)

$$F_{27}(x) = \sum_{i=1}^{6} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{27}^*$$

$\sigma = [10, 20, 30, 40, 50, 60],$
$\lambda = [10, 10, 2.5, 1e-26, 1e-6, 5e-4],$
$bias = [0, 100, 200, 300, 400, 500],$
$g = [f_{18}', f_5', f_{10}', f_{11}', f_{11}', f_6']$

Composition Function 8($F_{28}$)

$$F_{28}(x) = \sum_{i=1}^{6} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{28}^*$$

$\sigma = [10, 20, 30, 40, 50, 60], \lambda = [10, 10, 1e-6, 1, 1, 5e-4],$
$bias = [0, 100, 200, 300, 400, 500],$
$g = [f_{13}', f_{15}', f_{12}', f_4', f_{17}', f_6']$

Composition Function 9($F_{29}$)

$$F_{29}(x) = \sum_{i=1}^{3} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{29}^*$$

$\sigma = [10, 30, 50], \lambda = [1, 1, 1], bias = [0, 100, 200],$
$g = [F_5', F_6', F_7']$

Composition Function 10($F_{30}$)

$$F_{30}(x) = \sum_{i=1}^{3} \{\omega_i^*[\lambda_i g_i(x) + bias_i]\} + F_{30}^*$$

$\sigma = [10, 30, 50], \lambda = [1, 1, 1], bias = [0, 100, 200],$
$g = [F_5', F_8', F_9']$

Bent Cigar Function ($f_1$)

$$f_1(x) = x_1^2 + 10^6 \sum_{i=1}^{D} x_i^2$$

Sum of Different Power Function ($f_2$)

$$f_2(x) = \sum_{i=1}^{D} \mid (x_i) \mid^{i+1}$$

Zakharov Function ($f_3$)

$$f_3(x) = \sum_{i=1}^{D} x_i^2 + \left(0.5 \sum_{i=1}^{D} x_i\right)^2 + \left(0.5 \sum_{i=1}^{D} x_i\right)^4$$

Rosenbrock's Function ($f_4$)

$$f_4(x) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right]$$

Rastrigin's Function ($f_5$)

$$f_5(x) = \sum_{i=1}^{D} \left[x_i^2 - 10cos(2\pi x_i) + 10\right]$$

Expanded Schaffer's F6 Function ($f_6$)

$$f_6(x) = g(x_1, x_2) + g(x_2, x_3) + g(x_3, x_4) + \ldots + g(x_D, x_1)$$

$$g(x, y) = 0.5 + \frac{sin^2\left(\sqrt{x^2 + y^2}\right) - 0.5}{\left(1 + 0.001(x^2 + y^2)\right)^2}$$

Lunacek bi-Rastrigin Function ($f_7$)

$$f_7(x) = min\left(\sum_{i=1}^{D}\left(x_i^2\right) - \mu_0\right)^2, dD + S\sum_{i=1}^{D}(x_i - \mu_i)^2 + 10\left(D - \sum_{i=1}^{D} cos(2\pi z)\right)$$

Non-continuous Rotated Rastrigin's Function ($f_8$)

$$f_8(x) = \sum_{i=1}^{D}\left[x_i^2 - 10cos(2\pi x_i) + 10\right]$$

$$y_i = \begin{cases} x_i & |x_i| < 0.5 \\ \dfrac{round(2x_i)}{2} & |x_i| \geq 0.5 \end{cases}$$

Levy Function ($f_9$)

$$f_9(x) = sin^2(\pi\omega_i) + \sum_{i=1}^{D}[1 + 10sin^2(\pi\omega_i + 1)] + (\omega_D - 1)^2[1 + sin^2(1\pi\omega_D)]$$

Modified Schwefel's Function ($f_{10}$)

$$f_{10}(x) = 418.9829D - \sum_{i=1}^{D} g(z_i)$$

$$z_i = x_i + 4.209687462275036e + 002$$

High Conditioned Elliptic Function ($f_{11}$)

$$f_{11}(x) = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} x_i^2$$

Discus Function ($f_{12}$)

$$f_{12}(x) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$$

Ackley's Function ($f_{13}$)

$$f_{13}(x) = -20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2} - exp\left(\frac{1}{D}\sum_{i=1}^{D}cos(2\pi x_i)\right) + 20 + e\right)$$

Weierstrass Function ($f_{14}$)

$$f_{14}(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k20}\left[0.5^k cos(2\pi 3^k)(x_k + 0.5)\right]\right) - D\sum_{k=0}^{k20}\left[0.5^k cos(2\pi 3^k \cdot 0.5)\right]$$

Griewank's Function ($f_{15}$)

$$f_{15}(x) = \frac{1}{4000}\sum_{i=1}^{D}x_i^2 - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Katsuura Function ($f_{16}$)

$$f_{16}(x) = \frac{10}{D^2}\prod_{i=1}^{D}\left(1 + i\sum_{j=1}^{3} 2\frac{|\, 2^j x_i - round(2^j x_i)\,|}{2^j}\right)^{\frac{10}{D^{1.2}}} - \frac{10}{D}$$

HappyCat Function ($f_{17}$)

$$f_{17}(x) = |\left(\sum_{i=1}^{D}x_i^2\,|^{\frac{1}{4}} + \left(0.5\sum_{i=1}^{D}x_i^2 + \sum_{i=1}^{D}x_i\right)\right/D + 0.5$$

HGBat Function ($f_{18}$)

$$f_{18}(x) = |\left(\sum_{i=1}^{D}x_i^2\right)^2 - \left(\sum_{i=1}^{D}x_i\right)^2 |^{\frac{1}{2}} + \left(0.5\sum_{i=1}^{D}x_i^2 + \sum_{i=1}^{D}x_i\right)/D + 0.5$$

Expanded Griewank's plus Rosenbrock's Function ($f_{19}$)

$$f_{19}(x) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \ldots + f_7(f_4(x_D, x_1))$$

Schaffer's F7 Function ($f_{20}$)

$$f_{20}(x) = \left[\frac{1}{D-1}\left(\sum_{i=1}^{D-1}\left(\sqrt{s_i}(sin(50s_i^0.2) + 1)\right)\right]^2$$

$$s_i = \sqrt{x_i^2 + x_{i+1}^2}$$

## References


Agrawal, A., & Tripathi, S. (2019). Particle swarm optimization with probabilistic inertia weight. In *Harmony search and nature inspired optimization algorithms* (pp. 239–248). Springer.

Alswaitti, M., Albughdadi, M., & Isa, N. A. M. (2018). Density-based particle swarm optimization algorithm for data clustering. *Expert Systems with Applications, 91*, 170–186. doi:10.1016/j.eswa.2017.08.050.

Angeline, P. (1998). Using selection to improve particle swarm optimization. In *Proceedings of the 1998 ieee international conference on evolutionary computation, the 1998 ieee world congress on computational intelligence* (pp. 84–89). IEEE.

Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: A novel approach for global optimization. *Soft Computing, 23*(3), 715–734.

Awad, N., Ali, M., Liang, J., Qu, B., & Suganthan, P. (2016). Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. *Tech. Rep*.

Aydilek, I. B. (2018). A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Applied Soft Computing, 66*, 232–249.

Beheshti, Z., Shamsuddin, S. M., & Hasan, S. (2015). Memetic binary particle swarm optimization for discrete optimization problems. *Information Sciences, 299*, 58–84.

Carneiro, M. G., Cheng, R., Zhao, L., & Jin, Y. (2019). Particle swarm optimization for network-based data classification. *Neural Networks, 110*, 243–255.

Chen, K., Zhou, F., & Liu, A. (2018a). Chaotic dynamic weight particle swarm optimization for numerical function optimization. *Knowledge-Based Systems, 139*, 23–40.

Chen, Y., Li, L., Peng, H., Xiao, J., Yang, Y., & Shi, Y. (2017). Particle swarm optimizer with two differential mutation. *Applied Soft Computing, 61*, 314–330.

Chen, Y., Li, L., Xiao, J., Yang, Y., Liang, J., & Li, T. (2018b). Particle swarm optimizer with crossover operation. *Engineering Applications of Artificial Intelligence, 70*, 159–169.

Civicioglu, P. (2012). Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Computers & Geosciences, 46*, 229–247.

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation, 219*(15), 8121–8144.

Clerc, M. (2011). Standard particle swarm optimisation, from 2006 to 2011. *http://clerc.maurice.free.fr/pso/SPSO_descriptions.pdf (2011-07-13 version)*.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*, 3–18.

Doğan, B., & Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex search algorithm. *Information Sciences, 293*, 125–145.

Dong, W., Kang, L., & Zhang, W. (2017). Opposition-based particle swarm optimization with adaptive mutation strategy. *Soft Computing, 21*(17), 5081–5090.

Dosoglu, M. K., Guvenc, U., Duman, S., Sonmez, Y., & Kahraman, H. T. (2018). Symbiotic organisms search optimization algorithm for economic/emission dispatch problem in power systems. *Neural Computing and Applications, 29*(3), 721–737.

Eberhart, R., & Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the 2001 ieee congress on evolutionary computation: 1* (pp. 94–100). IEEE.

Eberhart, R. C., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 ieee congress on evolutionary computation* (pp. 84–88). IEEE.

El-Abd, M. (2013). Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks. In *Proceedings of the 2013 ieee congress on evolutionary computation* (pp. 2215–2220). IEEE.

Engelbrecht, A. P. (2005). *Fundamentals of computational swarm intelligence*: 1. Wiley Chichester.

Engelbrecht, A. P. (2016). Particle swarm optimization with crossover: A review and empirical analysis. *Artificial Intelligence Review, 45*(2), 131–165.

FarzanehKaloorazi, M., Bonev, I. A., & Birglen, L. (2018). Simultaneous path placement and trajectory planning optimization for a redundant coordinated robotic workcell. *Mechanism and Machine Theory, 130*, 346–362.

Geng, W. (2018). Cognitive deep neural networks prediction method for software fault tendency module based on bound particle swarm optimization. *Cognitive Systems Research, 52*, 12–20.

Hamdi, F., Ghorbel, A., Masmoudi, F., & Dupont, L. (2018). Optimization of a supply portfolio in the context of supply chain risk management: Literature review. *Journal of Intelligent Manufacturing, 29*(4), 763–788.

Han, H., Lu, W., & Qiao, J. (2017). An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. *IEEE Transactions on Cybernetics, 47*(9), 2754–2767.

Huang, J., Liu, W., Su, Y., & Wang, F. (2018). Cascade optimization design of inter-satellite link enhanced with adaptability in future GNSS satellite networks. *GPS Solutions, 22*(2), 44.

Jain, M., Maurya, S., Rani, A., & Singh, V. (2018). Owl search algorithm: A novel nature-inspired heuristic paradigm for global optimization. *Journal of Intelligent & Fuzzy Systems, 34*(3), 1573–1582.

Jindal, V., & Bedi, P. (2018). An improved hybrid ant particle optimization (IHAPO) algorithm for reducing travel time in VANETs. *Applied Soft Computing, 64*, 526–535.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm. *Journal of Global Optimization, 39*(3), 459–471.

Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 ieee congress on evolutionary computation: 3* (pp. 1931–1938). IEEE.

Kennedy, J. (2003). Bare bones particle swarms. In *Proceedings of the 2003 ieee swarm intelligence symposium* (pp. 80–87). IEEE.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the 1995 ieee international conference on neural networks: 4* (pp. 1942–1948). IEEE.

Kennedy, J., & Mendes, R. (2002). Population structure and particle swarm performance. In *Proceedings of the 2002 ieee congress on evolutionary computation: 2* (pp. 1671–1676). IEEE.

Krohling, R. A., & Mendel, E. (2013). Bare bones particle swarm optimization with gaussian or cauchy jumps. In *Proceedings of the 2009 ieee congress on evolutionary computation* (pp. 3285–3291). IEEE.

Lai, K., Illindala, M., & Subramaniam, K. (2019). A tri-level optimization model to mitigate coordinated attacks on electric power systems in a cyber-physical environment. *Applied Energy, 235*, 204–218.

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation, 10*(3), 281–295.

Liang, J. J., & Suganthan, P. N. (2005). Dynamic multi-swarm particle swarm optimizer with local search. In *Proceedings of the 2005 ieee congress on evolutionary computation: 1* (pp. 522–528). IEEE.

Lin, A., Sun, W., Yu, H., Wu, G., & Tang, H. (2019a). Global genetic learning particle swarm optimization with diversity enhancement by ring topology. *Swarm and Evolutionary Computation, 44*, 571–583.

Lin, G., Guan, J., Li, Z., & Feng, H. (2019b). A hybrid binary particle swarm optimization with tabu search for the set-union knapsack problem. *Expert Systems with Applications, 135*, 201–211. doi:10.1016/j.eswa.2019.06.007.

Liu, H., Ding, G., & Wang, B. (2014). Bare-bones particle swarm optimization with disruption operator. *Applied Mathematics and Computation, 238*, 106–122.

Liu, H., Xu, G., Ding, G., & Li, D. (2015). Integrating opposition-based learning into the evolution equation of bare-bones particle swarm optimization. *Soft Computing, 19*(10), 2813–2836. doi:10.1007/s00500-014-1444-0.

Lynn, N., & Suganthan, P. N. (2015). Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm and Evolutionary Computation, 24*, 11–24.

Mahmoodian, V., Jabbarzadeh, A., Rezazadeh, H., & Barzinpour, F. (2019). A novel intelligent particle swarm optimization algorithm for solving cell formation problem. *Neural Computing and Applications, 31*(2), 801–815.

Marinakis, Y., Marinaki, M., & Migdalas, A. (2019). A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows. *Information Sciences, 481*, 311–329.

Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation, 8*(3), 204–210.

Nagra, A. A., Han, F., Ling, Q., & Mehta, S. (2019a). An improved hybrid method combining gravitational search algorithm with dynamic multi swarm particle swarm optimization. *IEEE Access, 7*, 50388–50399. doi:10.1109/ACCESS.2019.2903137.

Nagra, A. A., Han, F., & Ling, Q. H. (2019b). An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search. *Engineering Optimization, 51*(7), 1115–1132.

Nakano, S., Ishigame, A., & Yasuda, K. (2010). Consideration of particle swarm optimization combined with tabu search. *Electrical Engineering in Japan, 172*(4), 31–37.

Nasir, M., Das, S., Maity, D., Sengupta, S., Halder, U., & Suganthan, P. N. (2012). A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Information Sciences, 209*, 16–36.

Nouiri, M., Bekrar, A., Jemai, A., Niar, S., & Ammari, A. C. (2018). An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *Journal of Intelligent Manufacturing, 29*(3), 603–615.

Omran, M. G., Engelbrecht, A. P., & Salman, A. (2009). Bare bones differential evolution. *European Journal of Operational Research, 196*(1), 128–139.

Peram, T., Veeramachaneni, K., & Mohan, C. K. (2003). Fitness-distance-ratio based particle swarm optimization. In *Proceedings of the 2003 ieee swarm intelligence symposium* (pp. 174–181). IEEE.

Raineri, M., Ronchini, F., Perri, S., & Bianco, C. G. L. (2020). Optimality criteria for the path planning of autonomous industrial vehicles. In *Advances in robotics research: From lab to market* (pp. 125–140). Springer.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). Gsa: A gravitational search algorithm. *Information sciences, 179*(13), 2232–2248.

Ratnaweera. , A., & Halgamuge. , S. K. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation, 8*(3), 240–255.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of the 1998 ieee world congress on computational intelligence, the 1998 ieee international conference on evolutionary computation* (pp. 69–73). IEEE.

Shi, Y., & Eberhart, R. (2001). Fuzzy adaptive particle swarm optimization. In *Proceedings of the 2001 congress on evolutionary computation: 1* (pp. 101–106). IEEE.

Tharwat, A., Elhoseny, M., Hassanien, A. E., Gabel, T., & Kumar, A. (2019). Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm. *Cluster Computing, 22*(2), 4745–4766.

Thurner, L., Scheidler, A., Schäfer, F., Menke, J.-H., Dollichon, J., Meier, F., … Braun, M. (2018). Pandapower–an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems, 33*(6), 6510–6521.

Vafashoar, R., & Meybodi, M. R. (2019). Cellular learning automata based bare bones PSO with maximum likelihood rotated mutations. *Swarm and Evolutionary Computation, 44*, 680–694. doi:10.1016/j.swevo.2018.08.016.

Wang, F., Zhang, H., Li, K. S., Lin, Z. Y., Yang, J., & Shen, X. L. (2018). A hybrid particle swarm optimization algorithm using adaptive learning strategy. *Information Sciences, 436*, 162–177.

Wang, S., Li, Y., & Yang, H. (2019). Self-adaptive mutation differential evolution algorithm based on particle swarm optimization. *Applied Soft Computing, 81*, 105496. doi:10.1016/j.asoc.2019.105496.

Wang, Y., Liu, H., Yu, Z., & Tu, L. (2020). An improved artificial neural network based on human-behaviour particle swarm optimization and cellular automata. *Expert Systems with Applications, 140*, 112862.

Wu, Q., Ma, Z., Fan, J., Xu, G., & Shen, Y. (2019a). A feature selection method based on hybrid improved binary quantum particle swarm optimization. *IEEE Access, 7*, 80588–80601.

Wu, Y., Wu, Y., & Liu, X. (2019b). Couple-based particle swarm optimization for short-term hydrothermal scheduling. *Applied Soft Computing, 74*, 440–450.

Xu, G., Cui, Q., Shi, X., Ge, H., Zhan, Z.-H., Lee, H. P., … Wu, C. (2019). Particle swarm optimization based on dimensional learning strategy. *Swarm and Evolutionary Computation, 45*, 33–51.

Xu, X., Rong, H., Trovati, M., Liptrott, M., & Bessis, N. (2018). Cs-pso: Chaotic particle swarm optimization algorithm for solving combinatorial optimization problems. *Soft Computing, 22*(3), 783–795.

Xue, Y., Tang, T., & Liu, A. X. (2019). Large-scale feedforward neural network optimization by a self-adaptive strategy and parameter based particle swarm optimization. *IEEE Access, 7*, 52473–52483.

Yang, C., Gao, W., Liu, N., & Song, C. (2015). Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight. *Applied Soft Computing, 29*, 386–394.

Yang, X.-S., & Deb, S. (2014). Cuckoo search: Recent advances and applications. *Neural Computing and Applications, 24*(1), 169–174.

Yu, H., Tan, Y., Zeng, J., Sun, C., & Jin, Y. (2018). Surrogate-assisted hierarchical particle swarm optimization. *Information Sciences, 454*, 59–72.

Zhang, K., Huang, Q., & Zhang, Y. (2019a). Enhancing comprehensive learning particle swarm optimization with local optima topology. *Information Sciences, 471*, 1–18.

Zhang, N., Li, C., Li, R., Lai, X., & Zhang, Y. (2016). A mixed-strategy based gravitational search algorithm for parameter identification of hydraulic turbine governing system. *Knowledge-Based Systems, 109*, 218–237.

Zhang, X., Liu, H., Zhang, T., Wang, Q., Wang, Y., & Tu, L. (2019b). Terminal crossover and steering-based particle swarm optimization algorithm with disturbance. *Applied Soft Computing, 85*, 105841.

Zhang, X., Wang, X., Kang, Q., & Cheng, J. (2019c). Differential mutation and novel social learning particle swarm optimization algorithm. *Information Sciences, 480*, 109–129.