Ensemble Strategy Using Particle Swarm Optimisation Variant and Enhanced Local Search Capability

Libin Hong[a,*], Guodong Wang[a], Ender Özcan[b], John Woodward[c]

[a]*School of Information Science and Technology, Hangzhou Normal University, No.2318 Yuhangtang Road, Hangzhou 31121, P.R.China*
[b]*School of Computer Science, University of Nottingham, Wollaton Road, Nottingham NG8 1BB, United Kingdom*
[c]*Department of Computer Science, Loughborough University, Epinal Way, Loughborough LE11 3TU, United Kingdom*

Abstract

Particle swarm optimisation is a population-based algorithm for evolutionary computation. A notable recent research direction has been to combine different effective mechanisms to enhance both exploration and exploitation capabilities while employing suitable mechanisms at appropriate instances in the evolutionary process. This study entailed the development of an ensemble strategy that uses a variant of the modified particle swarm optimisation algorithm with a covariance matrix adapted to the retreat phase and sequential quadratic programming. The modified particle swarm optimisation algorithm employs nonlinear population size reduction and uses the candidate elite best solution as the stochastic learning strategy and the fitness-distance balance as the terminal updating mechanism. The proposed algorithm was compared with the most recently proposed particle swarm optimisation-based variants through testing on CEC2017 benchmark functions. In the experimental results, the proposed method achieved the best ranking and exhibited excellent performance. Further effectiveness tests demonstrated that the proposed combination of algorithms and mechanisms exhibits tacit cooperation and significantly improves performance.

Keywords: Ensemble Strategy; Particle Swarm Optimisation; Covariance Matrix Adapted Retreat; Sequential Quadratic Programming; Fitness-Distance Balance;

*Corresponding author: Libin Hong. E-mail: libin.hong@hznu.edu.cn.

1. Introduction

The ensemble strategy produces excellent performance in evolutionary computation [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. The underlying concept of the ensemble strategy is to combine algorithms with different characteristics to utilise their advantages and to employ particular methods or mechanisms to further enhance both exploration and exploitation capabilities. The sophisticated framework proposed by Elsayed et al. [2, 3] demonstrated outstanding performance in solving complex optimisation problems. The primary idea underlying their effort was to design an ensemble strategy with subpopulations and apply a local search method to enhance the exploitation capability in later generations of evolution. Kumar et al. [4] proposed an effective butterfly optimiser using this framework that was one of the best performing algorithms in the CEC2017 competition. Hong et al. [9] proposed and utilised a variant of the united multi-operator evolutionary algorithms (UOMEAs), which they called UOMEAs-III, using improved success-history based parameter adaptation for differential evolution with ensemble sinusoidal approach and the covariance matrix learning for the crossover operator based on the Euclidean neighborhood (SHADE-cnEpSin) [11] and sequential quadratic programming (SQP). Yang et al. [12] proposed adaptive multistrategy ensemble particle swarm optimisation (AMSEPSO) and developed a stochastic elite-learning method for its application. Kahraman et al. [13] proposed a selection method based on the fitness-distance balance (FDB) mechanism to solve the premature convergence problem in the meta-heuristic search (MHS) process. In the CEC2017 [14] competition on single-objective real-parameter numerical optimization, the top-performing meta-heuristics were none PSO-based algorithms. This motivated us to design a PSO-based ensemble strategy to further enhance the performance. Inspired by this body of research, this study considered the following research objectives:

- *Further improving the sophisticated framework and integrating a PSO-based variant into it, which has been shown to be a robust and efficient approach.*
- *Combining the most recently proposed mechanisms, which can effectively control population diversity and balance global and local searches, with the PSO-based variant to further promote performance.*

PSO has been applied for image classification [15], diagnosing plant diseases [16], COVID-19 prediction [17], reliability engineering [18], the DNA fragment assembly problem [19], area coverage of heterogeneous sensors [20], complex manufacturing processes [21], and adversarial text attacks [22], among others. Due to its widespread use in real-world applications, the research on PSO-based variants has maintained a certain degree of interest in recent years. Zhang et al. [23] proposed a terminal crossover and steering-based PSO with distribution (TCSPSO) to overcome deterioration in population diversity. Liu et al. [24] proposed a modified particle swarm optimisation (MPSO) using an adaptive strategy and chaos-based nonlinear inertia weights. A pyramid PSO (PPSO) with competitive and cooperative mechanisms for renewing particle information was proposed by Li et al. [25]. Meng et al. [26] proposed a PSO-based single-objective numerical optimisation (PSO-sono) variant comprising an ensemble of inertia weight PSO [27] and social learning PSO [28]. Zhao and Wang [29] proposed an elite-ordinary synergistic PSO (EOPSO) method in which particles are divided into elite and ordinary members based on their fitness values. Further, Zhang [30] reported an elite archive-driven PSO (EAPSO) that utilises both population size and terminal conditions to perform the search. These PSO-based variants have led to significant improvements in terms of performance. The algorithm proposed in this paper has the following salient features:

- *A further improved MPSO (IMPSO) is proposed and integrated into a general sophisticated framework that uses the SQP method in the later generations of evolution; its effectiveness is verified via numerous experiments.*
- *The IMPSO is combined with the covariance matrix-adapted retreat phase (CMAR); a novel probabilities calculation method for both IMPSO and CMAR is proposed for the general sophisticated framework.*
- *The IMPSO employs the nonlinear population size reduction (NPSR) method to dynamically tune the subpopulation size, the candidate elite best solution for stochastic learning strategy, and the FDB mechanism to avoid premature convergence.*

The remainder of this paper is organised as follows. Section 2 introduces the general sophisticated framework used in this study and describes improvements to the framework in detail. Section 3 presents the variant of MPSO, introduces the mechanisms used by IMPSO, and presents the dynamic adjustment of the population size, stochastic learning strategy, and terminal replacement strategy for IMPSO. Section 4 presents the parameter settings for all PSO-based variants compared in this study and the experimental results. Comparisons, analysis, and discussions are also presented in this section. Finally, Section 5 summarises the study and discusses future work.

2. General Sophisticated Framework

The general sophisticated framework is presented in Algorithm 1, in which PS_{GSA} represents the population, which is also a subpopulation of the framework of a certain global search algorithm (GSA). In Elsayed et al. [2], two variants of the differential evolution (DE) algorithm were combined in a general framework without using CMAR. The GSA in [3] was a multi-operator differential evolution (MODE) algorithm, and CMAR was introduced in [31]. In the study by Kumar et al. [4], the GSA was shown to be an effective butterfly optimiser. The GSA introduced in [9], in which the SQP method was introduced, was an improved version of SHADE-cnEpSin [32]. In this paper, the population PS_{GSA} is the IMPSO subpopulation and is referred to as PS_{IMPSO}. The proposed algorithm is called IMPSOwithCMAR based on its utilisation of CMAR subpopulation PS_{CMAR}. Further, $Prob_{IMPSO}$ and $Prob_{CMAR}$ denote IMPSO and CMAR probabilities, respectively.

2.1. Calculating Probabilities $Prob_{IMPSO}$ and $Prob_{CMAR}$

In the original framework, $Prob_i$ was calculated using Equations 1, 2, 3, and 4:

$$\hat{Q}_i = \frac{f^{best}_{CS,i}}{f^{best}_{CS,IMPSO} + f^{best}_{CS,CMAR}}, i \in \{1, 2\} \tag{1}$$

where CS is the cycle value and is set to 50, 100, 150, and 150 for 10, 30, 50, and 100 dimensions, respectively, and $f^{best}_{CS,i}$ is the best fitness value at the end of the cycle from the i^{th} subpopulation (either IMPSO or CMAR subpopulation). The normalised diversity $\hat{div}_i$ is calculated using Equation 2:

$$\hat{div}_i = \frac{div_i}{div_1 + div_2}, i \in \{1, 2\} \tag{2}$$

$$PI_i = (1 - \hat{Q}_i) + \hat{div}_i, i \in \{1, 2\} \tag{3}$$

$$Prob_i = max(0.1, min(0.9, \frac{PI_i}{PI_1 + PI_2})), i \in \{1, 2\} \tag{4}$$

In this study, Equation 5 was used instead of Equation 1. Thus, Equations 2, 3, 4, and 5 were used to calculate $Prob_i$:

$$\hat{Q}_i = \frac{f^{rand}_{CS,i}}{f^{rand}_{CS,IMPSO} + f^{rand}_{CS,CMAR}}, i \in \{1, 2\} \tag{5}$$

where $f^{rand}_{CS,i}$ is the fitness value randomly selected from the IMPSO and CMAR subpopulations at the end of the cycle. The proposed method can slightly increase $Prob_{CMAR}$ and accumulate advantage over all generations; thus, the local search is more sufficient during the evolution.

Algorithm 1 General sophisticated framework [2, 3, 4, 9]

```
1: Define $PS = PS_{GSA} + PS_{CMAR}$, $Prob_{GSA} = Prob_{CMAR} = 1$, $Prob_{ls} = 0.1$, $cy = 0$ and all other parameters required
2: for $i : PS$ do
3:     $p_i \leftarrow$ uniformly distributed D random numbers
4: end for
5: while termination condition is not satisfied do
6:     $cy = cy + 1$
7:     if $cy == CS$ then
8:         Calculate $Prob_{GSA}$, $Prob_{CMAR}$
9:     end if
10:    if $cy == 2 \times CS$ then
11:        Share information
12:        $Prob_{GSA} = Prob_{CMAR} = 1$
13:        $cy = 0$
14:    end if
15:    if $rand(0,1) \leq Prob_{GSA}$ then
16:        Apply global search algorithm (GSA), update $cfe$, and sort $PS_{GSA}$
17:    end if
18:    if $rand(0,1) \leq Prob_{CMAR}$ then
19:        Apply CMA-ES, update $cfe$, and sort $PS_{CMAR}$
20:    end if
21:    if $cfe \geq 0.75 \times FEs_{max}$ then
22:        if rand(0,1) $\leq Prob_{ls}$ then
23:            Apply the local search (sequential quadratic programming) method to the best particle/individual to find a better solution
24:            if the solution is improved then
25:                $Prob_{ls} = 0.1$
26:                Replace the best solution in $PS_{GSA}$, $\vec{p}_{best,g}$ in $PS_{CMAR}$ and $\sigma$
27:            else
28:                $Prob_{ls} = 0.01$
29:            end if
30:        end if
31:    end if
32:    $g = g + 1$
33: end while
```

2.2. Covariance Matrix-Adapted Retreat Phase

The CMAR phase includes the following steps [31, 33, 34]:

- *Step01: $MN1(m, C)$ and $MN2(m, C)$, with the same probability, are two multivariate normal distributions used to generate new particles. The distributions are individually controlled by the mean $m \in \mathbb{R}^n$ and a symmetric and positive definite covariance matrix $C \in \mathbb{R}^{n\times n}$. Distributions $MN1$ and $MN2$ can be indicated as $MN1(m, C) \sim m + MN1(0, C)$ and $MN2(m, C) \sim m + MN2(0, C)$, respectively, where "$\sim$" denotes denotes similar in distribution.*

- *Step02: New particles are produced using the following equation:*

$$CMAR\_X_i^{g+1} = \begin{cases} m^g + \sigma^g \cdot MN1(0, C^g), & if \quad rand_i \leq 0.5 \\ m^g + \sigma^g \cdot MN2(0, C^g), & otherwise. \end{cases} \tag{6}$$

  where σ is a step size for restricting the range of the distribution and $CMAR\_X_i$ represents a particle from the CMAR subpopulation.

- *Step03: The mean m is evaluated on the basis of the weighted average of 50% of the best particles from the CMAR subpopulation ($\{CMAR\_X_1, CMAR\_X_2, CMAR\_X_3, ..., CMAR\_X_{PS_{CMAR}}\}$), as shown in Equation (7).*

$$m^g = \sum_{i=1}^{\frac{PS_{CMAR}}{2}} w_i^g \cdot CMAR\_X_i^g \tag{7}$$

  where PS_{CMAR} is the population size of the CMAR subpopulation and the weights w_i ($\sum_{i=1}^{\frac{PS_{CMAR}}{2}} w_i = 1$ and $w_1 \geq w_2 \geq ...w_{\frac{PS_{CMAR}}{2}} \geq 0$) are evaluated using Equation (8):

$$w_i^g = 1 - \frac{fit(CMAR\_X_i^g)}{\sum_{z=1}^{\frac{PS_{CMAR}}{2}} fit(CMAR\_X_z^g)} \tag{8}$$

- *Step04: The covariance matrix C^{g+1} and global step size σ^{g+1} adaption are evaluated using Equations (9) and (10), respectively, according to [31, 34]:*

$$C^{g+1} = (1 - c_1 - c_\mu) \cdot C^g + c_1 \cdot p_c^{g+1} \cdot (p_c^{g+1})^T + c_\mu \cdot \sum_{i=1}^{PS_{CMAR}} w_i \cdot y_i^{g+1} \cdot (y_i^{g+1})^T \tag{9}$$

  where p_c^{g+1} is the evolution path and $y_i^{g+1} = \frac{CMAR\_X_i^{g+1} - m^g}{\sigma^g}$; $c_1 = \frac{2}{(n+1.3)^2 + \mu_{eff}}$, $c_\mu = min(1 - c_1, 2 \cdot \frac{\mu_{eff} - 2 + \frac{1}{\mu_{eff}}}{(n+2)^2 + \mu_{eff}})$, μ_{eff} is the variance effective selection mass for the mean, and *n is the dimension size [31, 34].*

$$\sigma^{g+1} = \sigma^g \cdot exp(\frac{c_\sigma}{d_\sigma} \cdot (\frac{||p_\sigma^{g+1}||}{E||N(0,1)||} - 1)) \tag{10}$$

where $E||N(0,1)||$ is the expected length of a $(0, I)$-normally distributed random vector, $c_\sigma = \frac{\mu_{eff}+2}{n+\mu_{eff}+3}$, and $d_\sigma \approx 1$ is the damping parameter [31].

2.3. Strategy Using Sequential Quadratic Programming

The underlying concept of the SQP method is to convert a nonlinear problem into a linear one [35, 36]. The principle of the SQP method involves determining an appropriate direction and modelling a quadratic optimal problem. The nonlinear optimisation problem can be represented as follows:

$$minimize\ f(x),\ such\ that \begin{cases} h(x) = 0 \\ g(x) \leq 0 \end{cases} \tag{11}$$

The Lagrangian of the above formulation can be expressed as follows:

$$L(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x) \tag{12}$$

where λ and μ are Lagrangian multipliers. The SQP method is an iterative operation that repeatably builds the problem for a given iteration x_k using a quadratic programming subproblem. Moreover, x_k is used to construct a new iteration x_{k+1}. The subproblem can be established by linearising the constraints of x_k and can be written as follows:

$$minimize\ f^{'}(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T Hf(x_k)(x - x_k),\ such\ that \begin{cases} h(x_k) + h^{'}(x_k)(x - x_k) = 0 \\ g(x_k) + g^{'}(x_k)(x - x_k) \leq 0 \end{cases} \tag{13}$$

where $Hf(x_k)$ is the Hessian of f at $x \in R^n$. Considering that the SQP method relies heavily on the initial estimate [37], the proposed IMPSOwithCMAR only applies the SQP method to the current global best particle. The SQP method is used to further enhance the local search capability in the later generations of the proposed algorithm, i.e., after 75% of the overall evolution.

3. Variant of Modified Particle Swarm Optimisation

In MPSO, a chaos-based nonlinear inertia weight is used to balance the global and local search capabilities of the PSO. In this PSO-based variant, stochastic and mainstream learning strategies are adopted to avoid premature convergence and an adaptive position-updating strategy and terminal replacement mechanism are used to enhance the search capability [24]. In this study, the MPSO was further constructed with complementary mechanisms to form an improved MPSO (IMPSO).

3.1. Fixed Population Size vs Nonlinear Population Size Reduction

In MPSO, the population size is a fixed value, which is set to 50 for all dimensions (10, 30, 50, and 100D). In IMPSO, the fixed population size is replaced by a nonlinear population size reduction using Equations 14 and 15:

$$NFE_r = \frac{cfe}{FEs_{max}} \tag{14}$$

where cfe is the value of the present function evaluations and FEs_{max} is the maximum value of the evaluations; thus, NFE_r represents the ratio of the present function evaluations.

$$PS_{IMPSO,g+1} = round((PS_{IMPSO,min} - PS_{IMPSO,max}) \cdot NFE_r^{1-NFE_r} + PS_{IMPSO,max}) \tag{15}$$

where $PS_{IMPSO,max}$ and $PS_{IMPSO,min}$ represent the maximum and minimum population sizes of IMPSO, respectively, and are set to 10 and 70, respectively.

3.2. Stochastic Learning Strategy

A stochastic learning strategy enables particles to learn from other outstanding particles in a population and can be used to guide more diverse particles. MPSO uses Equations 16 and 17 to carry out its stochastic learning strategy. At each generation, two different particles are randomly selected from the population and the better particle is considered a candidate best solution ($cPbest$). The current best particle ($Pbest_i$) with $cPbest$ then selects the best particles as the stochastic best solution ($sPbest$):

$$cPbest(g) = argmin\{fit(Pbest_a(g)), fit(Pbest_b(g))\}, a \neq b \in \{1, 2, ..., N\} \tag{16}$$

where N denotes the population size.

$$sPbest_i(g) = \begin{cases} cPbest(g), & fit(cPbest(g)) < fit(Pbest_i) \\ Pbest_i(g), & otherwise. \end{cases} \tag{17}$$

In the proposed IMPSO, Equations 18 and 19 are used in the stochastic learning strategy. Unlike MPSO, the top 20% of particles are screened out from the overall population as the elite collection.

Two particles are randomly selected from the elite collection, and the remaining operations follow the those of MPSO as described above.

$$cePbest(g) = argmin\{fit(cePbest_a(g)), fit(cePbest_b(g))\}, a \neq b \in \{1, 2, ..., EN\} \tag{18}$$

where $cePbest$ represents the $pbest$ elite particles and EN is the size of the elite collection.

$$sePbest_i(g) = \begin{cases} cePbest(g), & fit(cePbest(g)) < fit(Pbest_i) \\ Pbest_i(g), & otherwise. \end{cases} \tag{19}$$

$sePbest$ denotes a stochastic elite solution. Equations 20 and 21 are the velocity update equations for IMPSO. If Equation 20 is applied and the best fitness value cannot be updated for five generations, Equation 21 is used, and vice versa. Thus, the two velocity update equations are periodically switched:

$$V_i(g+1) = w(g) \cdot V_i(g) + c_1 \cdot r_1 \cdot (sPbest_i(g) - X_i(g)) + c_2 \cdot r_2 \cdot (mPbest_i(g) - X_i(g)) \tag{20}$$

$$V_i(g+1) = w(g) \cdot V_i(g) + c_1 \cdot r_1 \cdot (sePbest_i(g) - X_i(g)) + c_2 \cdot r_2 \cdot (mPbest_i(g) - X_i(g)) \tag{21}$$

where $c_1 = c_2 = 2$, r_1, and r_2 are uniform distributions in (0, 1).

3.3. Terminal Replacement Mechanism

In MPSO, the terminal update mechanism uses Equations 22, 23, and 24. Equation 22 represents the global worst particle, the crossover Equation 23 generates a new particle $Nbest$, and Equation 24 represents the particle with a small fitness value that is selected as $GWorst$ from Equations 22 and 23:

$$GWorst(g) = argmax\{fit(x_1(g)), fit(x_2(g), ..., fit(x_N(g))\} \tag{22}$$

$$Nbest(g) = Gbest(g) + rand \cdot (Pbest_j(g) - Pbest_k(g)), j \neq k \in \{1, 2, ..., N\} \tag{23}$$

$$GWorst(g) = \begin{cases} Nbest(g), & fit(Nbest(g)) < fit(GWorst(g)) \\ GWorst(g), & otherwise. \end{cases} \tag{24}$$

In the IMPSO, this selection strategy is replaced by the FDB selection method, which has been proven to be a more effective terminal updating mechanism. The motivation behind the FDB selection method is to evaluate the particles that make the most significant contributions to the evolutionary process [13]:

$$Distance_{p_i} = \sum_{m=1}^{D} |x_{p_i}^m - x_{Pbest}^m|,_{p_i} \neq_{Pbest}, i \in \{1, 2, ..., N\}, m \in \{1, 2, ..., D\} \tag{25}$$

where N denotes the population size, m is the m^{th} dimension, and D is the dimension size. $Distance_{p_i}$ is the sum of the distances between x_{p_i} and x_{Pbest} in all dimensions.

$$normF_{p_i} = 1 - \frac{fit_{pi} - BestFit}{WorstFit - BestFit}, i \in \{1, 2, ..., N\} \tag{26}$$

$$normF_{D_{pi}} = \frac{Distance_{pi} - Distance_{min}}{Distance_{max} - Distance_{min}}, i \in \{1, 2, ..., N\} \tag{27}$$

$$S_{FDB,P_i} = w \cdot normF_{p_i} + (1 - w) \cdot normF_{D_{pi}}, i \in \{1, 2, ..., N\} \tag{28}$$

where w is set to 0.5, fit_{pi} is the fitness value of the i^{th} particle, $BestFit$ and $WorstFit$ are the best and worst fitness values of the particles, respectively, $Distance_{min}$ and $Distance_{max}$ are the minimum and maximum $Distance_p$ values of the particles, respectively, and the particle with the maximum S_{FDB} is selected and used to replace $Gbest$ in Equation 23. The FDB method considers both the fitness values and positions of the best particles.

4. Performance Evaluation

To better observe and evaluate the performance of the proposed IMPSOwithCMAR, the most recently proposed state-of-the-art PSO-based variants—EAPSO [30], AMSEEPSO [12], EOPSO [29], PSO-sono [26], PPSO [25], MPSO [24], and TCSPSO [23]—were selected for comparison. All of the experimental results were evaluated using the CEC2017 benchmark functions listed in Table 1.

Table 1: CEC2017 benchmark functions.

Types	f_n	Benchmark Functions	Search Range	$F(x^*)f_{bias}$
Unimodal Functions	f_1	Shifted and Rotated Bent Cigar Function	$[-100, 100]^D$	100
	f_2	Shifted and Rotated Sum Diff Pow Function	$[-100, 100]^D$	200
	f_3	Shifted and Rotated Zakharov Function	$[-100, 100]^D$	300
Simple Multimodal Functions	f_4	Shifted and Rotated Rosenbrock's Function	$[-100, 100]^D$	400
	f_5	Shifted and Rotated Rastrigin's Function	$[-100, 100]^D$	500
	f_6	Shifted and Rotated Expanded Scaffer's F7 Function	$[-100, 100]^D$	600
	f_7	Shifted and Rotated Lunacek Bi_Rastrigin Function	$[-100, 100]^D$	700
	f_8	Shifted and Rotated Non-Continuous Rastrigin's Function	$[-100, 100]^D$	800
	f_9	Shifted and Rotated Levy Function	$[-100, 100]^D$	900
	f_{10}	Shifted and Rotated Schwefel's Function	$[-100, 100]^D$	1000
Hybrid Functions	f_{11}	Hybrid Function 1 (N=3)	$[-100, 100]^D$	1100
	f_{12}	Hybrid Function 2 (N=3)	$[-100, 100]^D$	1200
	f_{13}	Hybrid Function 3 (N=3)	$[-100, 100]^D$	1300
	f_{14}	Hybrid Function 4 (N=4)	$[-100, 100]^D$	1400

Types	f_n	Benchmark Functions	Search Range	$F(x^*)f_{bias}$
	f_{15}	Hybrid Function 5 (N=4)	$[-100, 100]^D$	1500
	f_{16}	Hybrid Function 6 (N=4)	$[-100, 100]^D$	1600
	f_{17}	Hybrid Function 7 (N=5)	$[-100, 100]^D$	1700
	f_{18}	Hybrid Function 8 (N=5)	$[-100, 100]^D$	1800
	f_{19}	Hybrid Function 9 (N=5)	$[-100, 100]^D$	1900
	f_{20}	Hybrid Function 10 (N=6)	$[-100, 100]^D$	2000
Composition Functions	f_{21}	Composition Function 1 (N=3)	$[-100, 100]^D$	2100
	f_{22}	Composition Function 2 (N=3)	$[-100, 100]^D$	2200
	f_{23}	Composition Function 3 (N=4)	$[-100, 100]^D$	2300
	f_{24}	Composition Function 4 (N=4)	$[-100, 100]^D$	2400
	f_{25}	Composition Function 5 (N=5)	$[-100, 100]^D$	2500
	f_{26}	Composition Function 6 (N=5)	$[-100, 100]^D$	2600
	f_{27}	Composition Function 7 (N=6)	$[-100, 100]^D$	2700
	f_{28}	Composition Function 8 (N=6)	$[-100, 100]^D$	2800
	f_{29}	Composition Function 9 (N=3)	$[-100, 100]^D$	2900
	f_{30}	Composition Function 10 (N=3)	$[-100, 100]^D$	3000

4.1. Parameter Settings

FEs_{max} was calculated as $10^4\times$ D for all tested PSO-based variants; thus, the values of FEs_{max} were set to 10×10^4, 30×10^4, 50×10^4, and 10×10^5 for 10, 30, 50, and 100D, respectively. The parameter settings of the proposed IMPSOwithCMAR algorithm and the algorithms with which it was compared, the PSO-based variants using default settings, are listed in Table 2. The recommended default settings, which had been tuned by the respective proposers, yielded the best performance for the state-of-the-art PSO-based variants.

Table 2: Parameter settings for the compared PSO-based variants.

Algorithm	Year	Default parameter settings
IMPSOwithCMAR	–	Initial subpopulation size for IMPSO = 70 (for 10, 30, 50, and 100D), $cw = 4 \times r \times (1 - r)$, r is a random value in (0,1), $w = 0.9 \rightarrow 0.4$, subpopulation size for CMAR $= 4 + floor(3 \times ln(D))$. The CS for 10, 30, 50, and 100D were set as 50, 100, 150, and 150, respectively.
EAPSO	2023	$pop\ size = 100$ (for 10, 30, 50, and 100D).
AMSEPSO	2022	$pop\ size = 40$ (for 10, 30, 50, and 100D), $c1 = 2.5 \rightarrow 1.5$, $c2 = 1.5 \rightarrow 2.5$, $w = 0.9 \rightarrow 0.4$, $\sigma = 0.3 \times pop\ size$, $stag\ max = 5$.
EOPSO	2022	$pop\ size = 40$ (for 10, 30, 50, and 100D), $c = 1.49445$, $Rmax = 0.5$, $Rmin = 0.4$, and $G = 7$.
PSO-sono	2022	$pop\ size = 100$ (for 10, 30, 50, and 100D), $r = 0.5$, $\epsilon = \frac{D}{PS} \times 0.01$, $w = 0.9 \rightarrow 0.4$.

Algorithm	Year	Default parameter settings
PPSO	2022	$pop\ size = 64$ (for 10, 30, 50, and 100D), ρ is set to 0.008, 0.02, 0.04 and 0.008 for 10, 30, 50, and 100D, respectively.
MPSO	2020	$pop\ size = 50$ (for 10, 30, 50, and 100D), r is a random value in (0,1), $cw = 4\times r\times(1-r)$, $w = 0.9 \rightarrow 0.4$.
TCSPSO	2019	$pop\ size = 50$ (for 10, 30, 50, and 100D), $c1 = c2 = 2$, $w = 0.9 \rightarrow 0.4$.

4.2. Experimental Results

To better review the results, the ranking statistics are listed in Table 3. The 1^{st} rankings of the minimum, mean, and median values were calculated for all dimensional experiments on all of the compared PSO-based variants. The highest 1^{st} ranking is indicated in boldface. The Friedman tests were performed and the ranks are given in Table 4. To observe the effectiveness of the proposed combination mechanisms, the single mechanism was validated using the following criteria (the comparative data are listed in Table 5):

- *Nonlinear population size reduction vs. fixed population size.*
- *Stochastic learning strategy: using both sePbest/sPbest periodically vs. using only sPbest.*
- *Terminal updating mechanism: FDB vs. Gbest.*
- *Using CMAR subpopulation vs. not using CMAR subpopulation.*
- *Randomly selecting fitness value from subpopulation vs. selecting best fitness value from subpopulation for calculating $Prob_i$.*

The mean/minimum/median values, standard deviations, and rankings for the 10-, 30-, 50-, and 100-dimensional experiments are listed in Tables 8, 9, 10, and 11, respectively. All of the statistical values are based on 51 runs, the best values and top rankings are in boldface. The two-sided Wilcoxon rank sum test (WRST) [38] was performed at the 0.05 significance level to compare IMPSOwithCMAR with EAPSO, AMSEPSO, EOPSO, PSO-sono, PPSO, MPSO, and TCSPSO. In the tables, '>', '=', and '<' indicate where IMPSOwithCMAR achieved significantly better, no statistically significant difference, and significantly worse results compared to the other PSO-based variants, respectively.

4.3. Comparison and Analysis

In Table 3, IMPSOwithCMAR has the most 1st rankings for mean (11, 17, 16, and 13 instances, respectively), minimum (15, 13, 14, and 13, respectively), and median (14, 15, 14, and 13, respectively) values in the 10-, 30-, 50-, and 100-dimensional experiments, respectively. In the 100-dimensional experiments, EOPSO tied with IMPSOwithCMAR in achieving the most 1st rankings for median values (13). The total number of 1st rankings achieved by IMPSOwithCMAR are 40, 45, 44, and 39 in the 10-, 30-, 50-, and 100-dimensional experiments, respectively, representing the best performance among all compared PSO-based variants. In Table 4, the Friedman tests on mean values with 30 CEC2017 benchmark functions show that IMPSOwithCMAR ranks highest across all dimensions. It is evident that IMPSOwithCMAR has the best performance among all the compared algorithms. To better review the rankings, the radar maps are given in Figure 5. The rankings of IMPSOwithCMAR for mean and minimum values are shown in red lines. The closer to the center of the circle, the better the ranking. Obviously, IMPSOwithCMAR has better rankings among all compared algorithms.

Table 3: Statistics of rankings for IMPSOwithCMAR, EAPSO, AMSEPSO, EOPSO, PSO-sono, PPSO, MPSO, and TCSPSO on CEC2017 benchmark functions.

Dim	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
$10D$	Mean [Rankings 1st]	**11**	5	8	5	6	3	4	2
	Min [Rankings 1st]	**15**	10	10	5	12	8	11	12
	Median [Rankings 1st]	**14**	7	10	6	9	6	8	6
	TOTAL [Rankings 1st]	**40**	22	28	16	27	17	23	20
$30D$	Mean [Rankings 1st]	**17**	3	2	7	0	2	0	0
	Min [Rankings 1st]	**13**	8	12	9	5	4	5	2
	Median [Rankings 1st]	**15**	4	5	9	1	3	1	1
	TOTAL [Rankings 1st]	**45**	15	19	25	6	9	6	3
$50D$	Mean [Rankings 1st]	**16**	2	0	9	0	3	0	0
	Min [Rankings 1st]	**14**	2	6	12	1	2	0	2
	Median [Rankings 1st]	**14**	4	2	10	0	3	0	0
	TOTAL [Rankings 1st]	**44**	8	8	31	1	8	0	2
$100D$	Mean [Rankings 1st]	**13**	1	1	11	1	2	1	0
	Min [Rankings 1st]	**13**	3	4	10	0	1	0	1
	Median [Rankings 1st]	**13**	2	1	**13**	0	1	0	0
	TOTAL [Rankings 1st]	**39**	6	6	34	1	4	1	1

Table 4: Rankings of the Friedman test for IMPSOwithCMAR, EAPSO, AMSEPSO, EOPSO, PSO-sono, PPSO, MPSO, and TCSPSO on CEC2017 benchmark functions.

Dim	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
$10D$	Friedman rank	**2.8500**	5.2833	3.5333	4.6667	3.8000	5.7000	4.7833	5.3833
	Rank	**1**	6	2	4	3	8	5	7

Dim	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
$30D$	Friedman rank	**2.0167**	5.1167	4.2667	3.0333	4.6000	4.1333	5.3333	7.5000
	Rank	**1**	6	4	2	5	3	7	8
$50D$	Friedman rank	**1.8333**	4.7667	4.1333	2.9667	4.7000	4.2000	5.9333	7.4667
	Rank	**1**	6	3	2	5	4	7	8
$100D$	Friedman rank	**2.2667**	4.1000	4.3000	2.7667	4.7667	4.2333	6.3333	7.2333
	Rank	**1**	3	5	2	6	4	7	8

In Table 5, IMPSOwithCMAR using nonlinear population size reduction performs better than IMPSOwithCMAR using fixed population size on $f_1 - f_3$, f_5, $f_9 - f_{11}$, $f_{13} - f_{14}$, $f_{16} - f_{18}$, f_{20}, $f_{23} - f_{25}$, and $f_{28} - f_{30}$; the total number of best mean values is 19, of which IMPSOwithCMAR using fixed population size achieves 12. IMPSOwithCMAR using both *sePest* and *sPbest* for stochastic learning performs better than IMPSOwithCMAR using only *sPbest* on f_1, $f_3 - f_{11}$, $f_{13} - f_{19}$, $f_{21} - f_{25}$, and $f_{27} - f_{29}$; the number of best mean values is 25, of which IMPSOwithCMAR utilising only *sPbest* for stochastic learning achieves only 7. IMPSOwithCMAR using FDB as the terminal updating mechanism performs better than IMPSOwithCMAR using *Gbest* on $f_1 - f_5$, $f_7 - f_{11}$, $f_{13} - f_{16}$, f_{18}, $f_{21} - f_{24}$, f_{27}, and $f_{29} - f_{30}$; the number of best mean values is 22, of which IMPSOwithCMAR utilising only *Gbest* as the terminal updating mechanism achieves 10. IMPSOwithCMAR with CMAR performs better than IMPSOwithCMAR without CMAR on $f_1 - f_5$, $f_7 - f_8$, f_{10}, $f_{13} - f_{14}$, $f_{16} - f_{18}$, $f_{20} - f_{21}$, and $f_{23} - f_{29}$; the number of best mean values is 22, of which IMPSOwithCMAR without CMAR achieves only 9. The IMPSOwithCMAR variant that randomly selects fitness values from subpopulations to calculate $Prob_i$ performs better than the IMPSOwithCMAR variant that selects the best fitness values on $f_1 - f_4$, $f_6 - f_7$, $f_9 - f_{11}$, f_{13}, $f_{15} - f_{17}$, f_{19}, and $f_{22} - f_{30}$; the number of best mean values is 23, of which IMPSOwithCMAR using best mean values achieves only 9.

In the 10-dimensional experiments (see Table 8) for the unimodal functions, IMPSOwithCMAR had the best minimum and median values on $f_1 - f_3$ and the best mean values on f_1 and f_3. On the simple multimodal functions, IMPSOwithCMAR had the best minimum and mean values on f_4 and $f_8 - f_9$ and the best median values on f_4 and f_9, whereas PPSO had the best minimum, mean, and median values on $f_5 - f_7$. EOPSO had the best mean and median values on f_{10}, whereas IMPSOwithCMAR, EOPSO, and TCSPSO had equivalent median values on f_{10}. On the hybrid functions, IMPSOwithCMAR had the best mean and median values on $f_{11} - f_{13}$ and the best minimum values on $f_{11} - f_{12}$; AMSEPSO, MPSO, and PSO-sono had the best minimum, mean, and median values on f_{14}, f_{15}, and f_{18}, respectively; AMSEPSO had the best mean values on $f_{16} - f_{17}$. PSO-sono and EOPSO had the best mean values on f_{19} and f_{20}, respectively. On the composition functions, IMPSOwithCMAR had the best minimum, mean, and median values on f_{24}, f_{26}, and f_{29}, respectively; EAPSO had the best minimum, mean, and median values on f_{21} as well as the best mean values on f_{28}; AMSEPSO had the best minimum, mean, and median values on $f_{22} - f_{23}$; AMSEPSO, PSO-sono, and EOPSO had the best mean values on f_{25}, f_{27}, and f_{30}, respectively. Generally, IMPSOwithCMAR performed better on unimodal functions in the low-dimensional experiments and exhibited excellent performance on some of the simple multimodal, hybrid, and composition functions.

In the 30-dimensional experiments (see Table 9) for the unimodal functions, IMPSOwithCMAR had the best minimum, mean, and median values on f_1 and f_3, whereas EAPSO had the best minimum, mean, and median values on f_2. On the simple multimodal functions, IMPSOwithCMAR had the best minimum, mean, and median values on f_4 and the best mean values on f_5 and f_8; EAPSO and EOPSO had the best mean values on f_6 and f_9, respectively; PPSO had the best mean values on f_7 and f_{10}. On the hybrid functions, IMPSOwithCMAR had the best mean values on $f_{11} - f_{15}$ and $f_{18} - f_{19}$; EOPSO had the best mean values on f_{16}; AMSEPSO had the best mean values on f_{17} and f_{20}. On the composition functions, IMPSOwithCMAR had the best minimum and median values on $f_{21} - f_{24}$, f_{26}, and f_{28} and the best mean values on f_{21}, f_{23}, f_{24}, f_{26}, and f_{28}; EOPSO had the best mean values on f_{25}, f_{27}, and $f_{29} - f_{30}$ as well as the best minimum and median values on f_{25}, f_{27}, and f_{30}. Generally, IMPSOwithCMAR performed better on the unimodal, simple multimodal, hybrid, and composition functions.

In the 50-dimensional experiments (see Table 10) for the unimodal functions, IMPSOwithCMAR had the best minimum, mean, and median values on f_1 and f_3, whereas EAPSO had the best mean and median values on f_2. On the simple multimodal functions, IMPSOwithCMAR had the best mean values on f_4, f_5, and f_8; EOPSO had the best mean values on f_6 and f_9; PPSO had the best mean values on f_7 and f_{10}. On the hybrid functions, IMPSOwithCMAR has the best minimum and median values on $f_{11} - f_{15}$ and $f_{18} - f_{19}$, the best mean values on $f_{12} - f_{15}$ and $f_{18} - f_{19}$, and the best minimum value on f_{19}; EOPSO had the best mean and median values on $f_{16} - f_{17}$ and f_{20}. On the composition functions, IMPSOwithCMAR had the best mean values on f_{21}, $f_{23} - f_{24}$, f_{26}, and f_{28}; PPSO had the best mean values on f_{22}; EOPSO had the best mean values on f_{25}, f_{27}, and $f_{29} - f_{30}$. Generally, IMPSOwithCMAR performed better on unimodal, simple multimodal, and hybrid functions. The performances of IMPSOwithCMAR and EOPSO were nearly equivalent on the composition functions.

In the 100-dimensional experiments (see Table 11) for the unimodal functions, IMPSOwithC-MAR had the best minimum, mean, and median values on f_1 and f_3, whereas MPSO had the best mean values on f_2. On the simple multimodal functions, IMPSOwithCMAR had the best minimum, mean, and median values on f_4; EOPSO had the best values on $f_5 - f_6$ and $f_8 - f_9$; and PSO-sono and PPSO had the best mean values on f_7 and f_{10}, respectively. On the hybrid functions, IMPSOwithCMAR had the best mean and median values on $f_{12} - f_{15}$ and $f_{18} - f_{19}$ and the best minimum values on f_{12},f_{14}, f_{15} and f_{18}; EAPSO had the best minimum, mean, and median values on f_{11}; EOPSO had the best minimum, mean, and median values on $f_{16} - f_{17}$ and f_{20}. On the composition functions, IMPSOwithCMAR had the best minimum, mean, and median values on $f_{24} - f_{26}$ and f_{28}; EOPSO had the best minimum, mean, and median values on f_{21}, f_{27}, and $f_{29} - f_{30}$; PPSO and AMSEPSO had the best mean values on f_{22} and f_{23}. Generally, IMPSOwithCMAR performed better on the unimodal and hybrid functions, EOPSO performed better on simple multimodal functions, and the performances of IMPSOwithCMAR and EOPSO on composition functions were nearly equivalent in the high-dimensional experiments.

The experimental results demonstrate that the proposed IMPSOwithCMAR algorithm exhibited much better performance in all dimensional experiments on the CEC2017 benchmark functions than the most recent PSO-based variants. This suggests that combining mechanisms using an ensemble strategy is generally effective and achieves significant improvement.

4.4. Search Behaviour of IMPSOwithCMAR

Figure 1 shows the evolutions of the best fitness value by source for various functions on single 50-dimensional runs. The search patterns are as follows:

- *The IMPSO subpopulation dominates effective evolution in the earlier generations and the CMAR subpopulation dominates effective evolution in the later generations.*
- *The IMPSO subpopulation dominates effective evolution in the earlier generations, CMAR dominates effective evolution in the middle generations, and the SQP method dominates effective evolution in the later generations.*

- *The IMPSO subpopulation dominates effective evolution in the earlier and middle generations; therefore, the SQP method dominates effective evolution in the later generations.*
- *The IMPSO subpopulation dominates effective evolution in the earlier and middle generations; therefore, the SQP method dominates on an instantaneous effective search and CMAR dominates effective evolution in the later generations.*

Figure 2 illustrates the differences between nonlinear population size reduction and fixed population size across all dimensions. The nonlinear population size reduction illustrates its effectiveness in the middle and later stages of evolution, as it further extends the maximum generation number of the evolutionary process when the FEs_{max} is fixed; thus, it contributes to a more effective and comprehensive evolution.

The two-dimensional particle distributions at various generations (0, 60, 120, and 240) obtained by applying IMPSOwithCMAR and IMPSOwithCMAR with different mechanisms on f_6 and f_{10} are plotted in Figures 3 and 4, respectively. The orange star represents the global optimum. The results indicate that the particles gather more during the evolutionary process when IMPSOwithCMAR utilises nonlinear population size reduction, as this reduction strategy can lead to a more sufficient evolution. In carrying out its stochastic learning strategy, IMPSOwithCMAR uses only $sPbest$, which causes the particles to become more scattered because using both $sePbest$ and $sPbest$ periodically weakens population diversity owing to elitism. IMPSOwithCMAR using FDB as the terminal updating mechanism causes the particles to scatter more in the later generations of evolution, as it considers both fitness value and distance to the best particle, it helps maintain better population diversity in the later generations. IMPSOwithCMAR without CMAR causes the particles to become more scattered during the evolutionary process, indicating that CMAR is an effective method for further enhancing local search capability. IMPSOwithCMAR uses random fitness values to calculate $Prob_i$ and causes the particles to gather more in both the earlier and later generations, which increases the utilization of CMAR. Thus, its exploitation is more sufficient for particles.

Convergence characteristics graphs of IMPSOwithCMAR and the PSO-based variants are presented in Figures 6 and 7. Owing to space limitations, only the evolutionary processes of the 50-dimensional experiments are presented. Figures 6 and 7 show the evolutionary processes of the PSO-based variants in the 50-dimensional experiments. Evidently, IMPSOwithCMAR produces a

Table 5: Mean best values and standard deviations, from 50-dimensional experiments, of IMPSOwithCMAR using different mechanisms.

f_n	IMPSOwithCMAR (Population Size Reduction)		IMPSOwithCMAR (Stochastic Learning Strategy)		IMPSOwithCMAR (Terminal Updating Mechanism)		IMPSOwithCMAR (CMAR Subpopulation)		IMPSOwithCMAR ($Prob_i$ Calculation)	
	Nonlinear	Fixed	$sePbest/sPbest$	$sPbest$	FDB	$Gbest$	CMAR	without CMAR	$rand$	$best$
f_1	**0.000000E+00**	1.960784E-07	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	1.450353E-02	**0.000000E+00**	**0.000000E+00**
	(0.000000E+00)	(1.400280E-06)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(4.011207E-03)	(0.000000E+00)	(0.000000E+00)
f_2	**5.074570E+20**	7.556474E+21	5.074570E+20	**3.843155E+20**	**5.074570E+20**	4.033130E+22	**5.074570E+20**	5.140242E+46	**5.074570E+20**	7.569425E+20
	(1.285972E+21)	(1.447603E+22)	(1.285972E+21)	(5.031772E+20)	(1.285972E+21)	(2.866603E+23)	(1.285972E+21)	(3.561557E+47)	(1.285972E+21)	(1.317884E+21)
f_3	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)
f_4	1.652989E+01	**1.436447E+01**	**1.652989E+01**	1.739267E+01	**1.652989E+01**	1.676030E+01	**1.652989E+01**	1.867852E+01	**1.652989E+01**	2.925816E+01
	(2.200168E+01)	(2.730786E+01)	(2.200168E+01)	(2.926256E+01)	(2.200168E+01)	(1.417270E+01)	(2.200168E+01)	(3.943715E+01)	(2.200168E+01)	(2.847363E+01)
f_5	**3.909602E+01**	3.973998E+01	**3.909602E+01**	8.059154E+01	**3.909602E+01**	1.257873E+02	**3.909602E+01**	6.821674E+01	3.909602E+01	**3.663789E+01**
	(6.384170E+00)	(9.067501E+00)	(6.384170E+00)	(1.776266E+01)	(6.384170E+00)	(4.782002E+01)	(6.384170E+00)	(2.401021E+01)	(6.384170E+00)	(4.953674E+00)
f_6	3.090204E+00	**2.761188E+00**	**3.090204E+00**	4.345277E+00	3.090204E+00	**2.815848E+00**	3.090204E+00	**2.675105E+00**	**3.090204E+00**	1.042741E+01
	(8.827571E-01)	(8.143522E-01)	(8.827571E-01)	(1.668538E+00)	(8.827571E-01)	(2.703916E+00)	(8.827571E-01)	(8.286614E-01)	(8.827571E-01)	(2.478042E+00)
f_7	1.205140E+02	**1.190375E+02**	**1.205140E+02**	1.357223E+02	**1.205140E+02**	1.950286E+02	**1.205140E+02**	1.302344E+02	**1.205140E+02**	1.933109E+02
	(1.605322E+01)	(1.527516E+01)	(1.605322E+01)	(1.903793E+01)	(1.605322E+01)	(3.855136E+01)	(1.605322E+01)	(2.644917E+01)	(1.605322E+01)	(3.123990E+01)
f_8	3.909601E+01	**3.817909E+01**	**3.909601E+01**	8.129387E+01	**3.909601E+01**	1.227698E+02	**3.909601E+01**	6.857306E+01	3.909601E+01	**3.661838E+01**
	(8.977350E+00)	(8.963369E+00)	(8.977350E+00)	(2.352837E+01)	(8.977350E+00)	(4.348351E+01)	(8.977350E+00)	(2.598921E+01)	(8.977350E+00)	(5.309829E+00)
f_9	**3.637730E+01**	3.950202E+01	**3.637730E+01**	1.961577E+02	**3.637730E+01**	2.408222E+02	3.637730E+01	**2.129060E+01**	**3.637730E+01**	4.522915E+02
	(4.014715E+01)	(3.887066E+01)	(4.014715E+01)	(2.299676E+02)	(4.014715E+01)	(1.807168E+02)	(4.014715E+01)	(1.602884E+01)	(4.014715E+01)	(2.482223E+02)
f_{10}	**4.786255E+03**	4.944079E+03	**4.786255E+03**	4.853783E+03	**4.786255E+03**	5.280389E+03	**4.786255E+03**	5.770018E+03	**4.786255E+03**	4.878652E+03
	(5.773028E+02)	(6.267532E+02)	(5.773028E+02)	(6.904134E+02)	(5.773028E+02)	(6.190256E+02)	(5.773028E+02)	(8.798105E+02)	(5.773028E+02)	(4.541634E+02)
f_{11}	**8.518573E+01**	9.715718E+01	**8.518573E+01**	1.228085E+02	**8.518573E+01**	1.410071E+02	8.518573E+01	**8.169243E+01**	**8.518573E+01**	1.710413E+02
	(3.095221E+01)	(3.203927E+01)	(3.095221E+01)	(3.457159E+01)	(3.095221E+01)	(5.340449E+01)	(3.095221E+01)	(2.236365E+01)	(3.095221E+01)	(4.080158E+01)
f_{12}	3.996475E+03	**2.192822E+03**	3.996475E+03	**3.090086E+03**	3.996475E+03	**3.034291E+03**	3.996475E+03	**2.175665E+03**	3.996475E+03	**3.969309E+03**
	(5.312949E+03)	(4.293706E+02)	(5.312949E+03)	(3.515533E+03)	(5.312949E+03)	(4.638244E+03)	(5.312949E+03)	(5.837183E+02)	(5.312949E+03)	(3.915211E+03)
f_{13}	**1.819712E+02**	2.187403E+02	**1.819712E+02**	5.083886E+02	**1.819712E+02**	2.289646E+02	**1.819712E+02**	2.774892E+02	**1.819712E+02**	5.617428E+02
	(1.153694E+02)	(1.448069E+02)	(1.153694E+02)	(2.069971E+02)	(1.153694E+02)	(1.590631E+02)	(1.153694E+02)	(2.072864E+02)	(1.153694E+02)	(2.448512E+02)
f_{14}	**2.320420E+02**	2.361286E+02	**2.320420E+02**	2.555836E+02	**2.320420E+02**	4.223272E+02	**2.320420E+02**	2.536970E+02	2.320420E+02	**2.317414E+02**
	(5.384711E+01)	(4.983018E+01)	(5.384711E+01)	(6.621083E+01)	(5.384711E+01)	(9.033503E+01)	(5.384711E+01)	(7.157658E+01)	(5.384711E+01)	(4.548500E+01)
f_{15}	6.508189E+01	**5.125247E+01**	**6.508189E+01**	2.235361E+02	**6.508189E+01**	1.033136E+02	6.508189E+01	**4.406010E+01**	**6.508189E+01**	3.501849E+02
	(9.129032E+01)	(3.582714E+01)	(9.129032E+01)	(1.194173E+02)	(9.129032E+01)	(1.032933E+02)	(9.129032E+01)	(1.912828E+01)	(9.129032E+01)	(1.599760E+02)
f_{16}	**8.844866E+02**	9.259190E+02	**8.844866E+02**	9.669544E+02	**8.844866E+02**	1.034616E+03	**8.844866E+02**	1.058205E+03	**8.844866E+02**	9.858559E+02
	(3.402577E+02)	(2.862129E+02)	(3.402577E+02)	(2.528168E+02)	(3.402577E+02)	(3.561590E+02)	(3.402577E+02)	(3.713760E+02)	(3.402577E+02)	(2.793661E+02)
f_{17}	**8.674605E+02**	1.004698E+03	**8.674605E+02**	9.034896E+02	8.674605E+02	**8.423370E+02**	**8.674605E+02**	1.114667E+03	**8.674605E+02**	9.209531E+02
	(2.487628E+02)	(2.573087E+02)	(2.487628E+02)	(2.289824E+02)	(2.487628E+02)	(2.103479E+02)	(2.487628E+02)	(2.752159E+02)	(2.487628E+02)	(2.555428E+02)
f_{18}	**2.020411E+02**	2.042487E+02	**2.020411E+02**	2.087521E+02	**2.020411E+02**	3.925938E+02	**2.020411E+02**	2.027430E+02	2.020411E+02	**2.019408E+02**
	(7.859069E+01)	(7.897510E+01)	(7.859069E+01)	(8.582360E+01)	(7.859069E+01)	(1.479432E+02)	(7.859069E+01)	(8.624597E+01)	(7.859069E+01)	(5.868318E+01)
f_{19}	2.764872E+02	**8.060342E+01**	**2.764872E+02**	3.079478E+02	2.764872E+02	**1.587068E+02**	2.764872E+02	**1.115424E+02**	**2.764872E+02**	4.084912E+02
	(2.449418E+02)	(8.520040E+01)	(2.449418E+02)	(2.661612E+02)	(2.449418E+02)	(1.468046E+02)	(2.449418E+02)	(1.496939E+02)	(2.449418E+02)	(3.179049E+02)
f_{20}	**6.208777E+02**	6.341689E+02	6.208777E+02	**5.781434E+02**	6.208777E+02	**6.159636E+02**	**6.208777E+02**	7.666881E+02	6.208777E+02	**6.153454E+02**
	(2.111587E+02)	(2.117222E+02)	(2.111587E+02)	(2.274219E+02)	(2.111587E+02)	(2.534433E+02)	(2.111587E+02)	(2.589130E+02)	(2.111587E+02)	(2.131801E+02)
f_{21}	2.425725E+02	**2.400664E+02**	**2.425725E+02**	2.826412E+02	**2.425725E+02**	3.095920E+02	**2.425725E+02**	2.644140E+02	2.425725E+02	**2.399031E+02**
	(9.485986E+00)	(8.315801E+00)	(9.485986E+00)	(2.423769E+01)	(9.485986E+00)	(2.695365E+01)	(9.485986E+00)	(2.133853E+01)	(9.485986E+00)	(6.536108E+00)
f_{22}	3.200991E+03	**2.578652E+03**	**3.200991E+03**	3.273726E+03	**3.200991E+03**	4.207408E+03	3.200991E+03	**2.788200E+03**	**3.200991E+03**	3.580946E+03
	(2.467572E+03)	(2.597898E+03)	(2.467572E+03)	(2.524613E+03)	(2.467572E+03)	(3.168536E+03)	(2.467572E+03)	(2.936282E+03)	(2.467572E+03)	(2.537579E+03)
f_{23}	**4.733980E+02**	4.747444E+02	**4.733980E+02**	5.519845E+02	**4.733980E+02**	5.441868E+02	**4.733980E+02**	5.031684E+02	**4.733980E+02**	4.792056E+02
	(1.268818E+01)	(1.540233E+01)	(1.268818E+01)	(4.691561E+01)	(1.268818E+01)	(4.100698E+01)	(1.268818E+01)	(3.061429E+01)	(1.268818E+01)	(1.297123E+01)
f_{24}	**5.340279E+02**	5.398171E+02	**5.340279E+02**	5.852659E+02	**5.340279E+02**	5.853579E+02	**5.340279E+02**	5.848359E+02	**5.340279E+02**	5.348065E+02
	(9.323682E+00)	(3.324679E+01)	(9.323682E+00)	(4.444704E+01)	(9.323682E+00)	(2.778552E+01)	(9.323682E+00)	(4.936690E+01)	(9.323682E+00)	(8.493858E+00)
f_{25}	**4.921153E+02**	5.202273E+02	**4.921153E+02**	4.966133E+02	4.921153E+02	**4.876028E+02**	**4.921153E+02**	5.581188E+02	**4.921153E+02**	4.944442E+02
	(2.995282E+01)	(4.205908E+01)	(2.995282E+01)	(3.426260E+01)	(2.995282E+01)	(3.416169E+01)	(2.995282E+01)	(3.182110E+01)	(2.995282E+01)	(2.267624E+01)
f_{26}	6.818628E+02	**6.419335E+02**	6.818628E+02	**4.973953E+02**	6.818628E+02	**4.751222E+02**	**6.818628E+02**	8.966273E+02	**6.818628E+02**	1.630136E+03
	(5.741020E+02)	(5.637867E+02)	(5.741020E+02)	(8.593668E+02)	(5.741020E+02)	(6.257920E+02)	(5.741020E+02)	(7.696392E+02)	(5.741020E+02)	(2.144477E+02)
f_{27}	5.689737E+02	**5.632963E+02**	**5.689737E+02**	6.799323E+02	**5.689737E+02**	6.227916E+02	**5.689737E+02**	5.728964E+02	**5.689737E+02**	6.385644E+02
	(3.936830E+01)	(3.076482E+01)	(3.936830E+01)	(6.980471E+01)	(3.936830E+01)	(7.012789E+01)	(3.936830E+01)	(3.752122E+01)	(3.936830E+01)	(4.013248E+01)
f_{28}	**4.660318E+02**	4.845199E+02	**4.660318E+02**	4.730957E+02	4.660318E+02	**4.657547E+02**	**4.660318E+02**	4.848000E+02	**4.660318E+02**	4.739826E+02
	(1.536594E+01)	(2.024521E+01)	(1.536594E+01)	(2.113445E+01)	(1.536594E+01)	(1.656620E+01)	(1.536594E+01)	(1.928149E+01)	(1.536594E+01)	(2.251776E+01)
f_{29}	**8.871036E+02**	9.414644E+02	**8.871036E+02**	9.828404E+02	**8.871036E+02**	9.641488E+02	**8.871036E+02**	1.017142E+03	**8.871036E+02**	1.081401E+03
	(1.880332E+02)	(2.645473E+02)	(1.880332E+02)	(2.737217E+02)	(1.880332E+02)	(2.504692E+02)	(1.880332E+02)	(3.083615E+02)	(1.880332E+02)	(2.048578E+02)
f_{30}	**5.994307E+05**	6.002623E+05	5.994307E+05	**5.983551E+05**	**5.994307E+05**	6.172914E+05	5.994307E+05	**5.991854E+05**	**5.994307E+05**	6.857332E+05
	(2.771337E+04)	(3.588133E+04)	(2.771337E+04)	(2.267990E+04)	(2.771337E+04)	(4.443954E+04)	(2.771337E+04)	(2.652987E+04)	(2.771337E+04)	(1.106726E+05)
Total	**19**	12	**25**	7	**22**	10	**22**	9	**23**	9

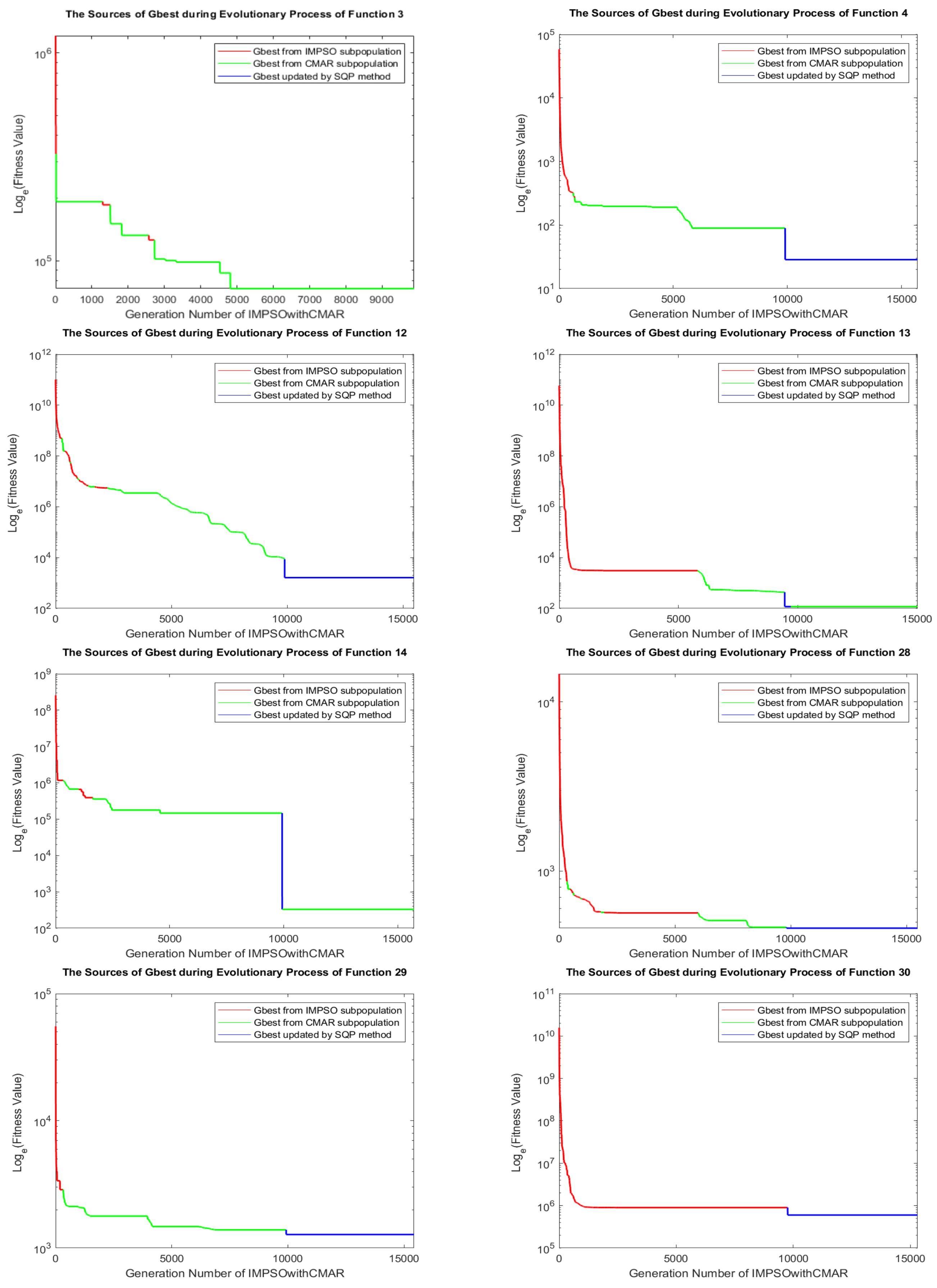

Figure 1: Sources of best particle in 50-dimensional experiments over the evolutionary process of an individual run.

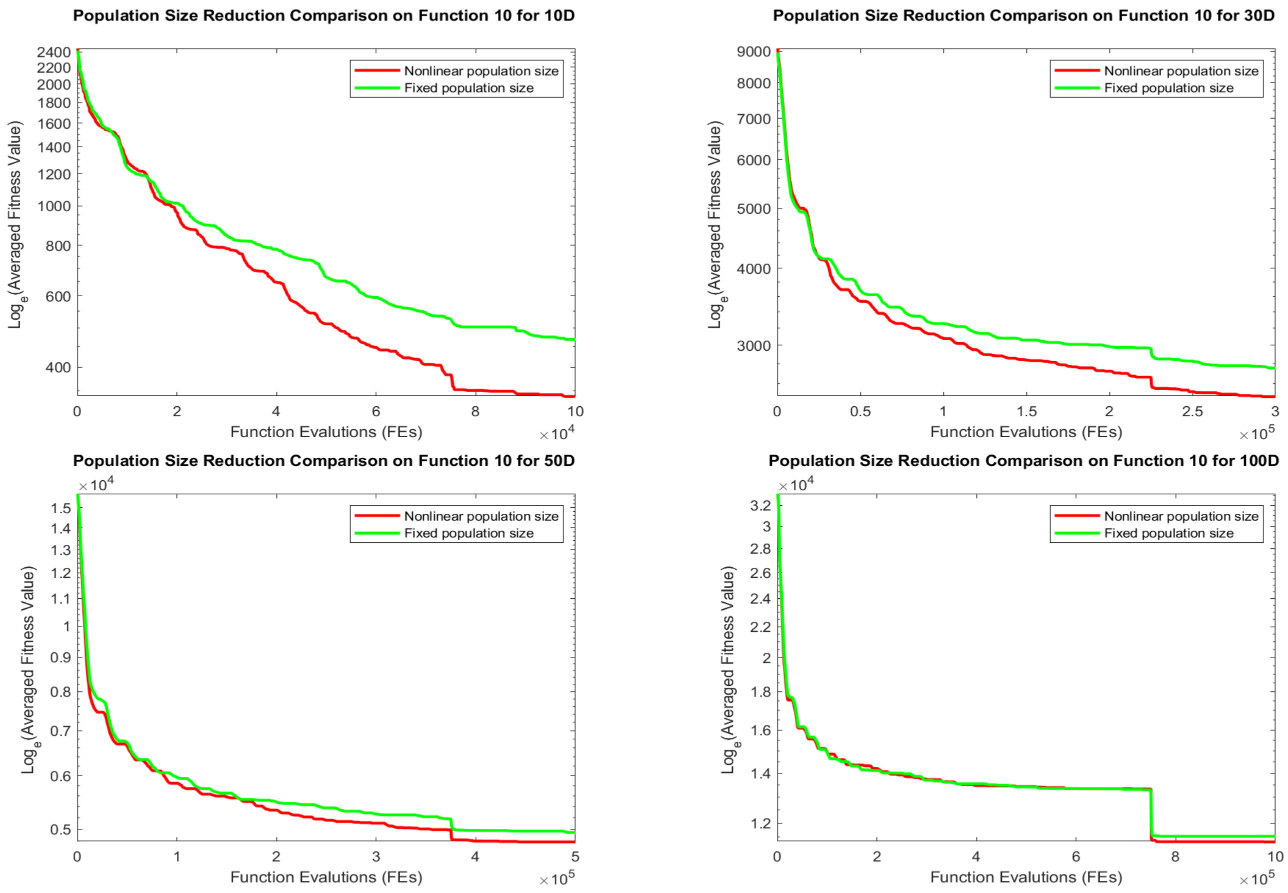

Figure 2: Comparison of population size reduction, averaged over 51 runs, for f_{10} across dimensions of 10, 30, 50, and 100.

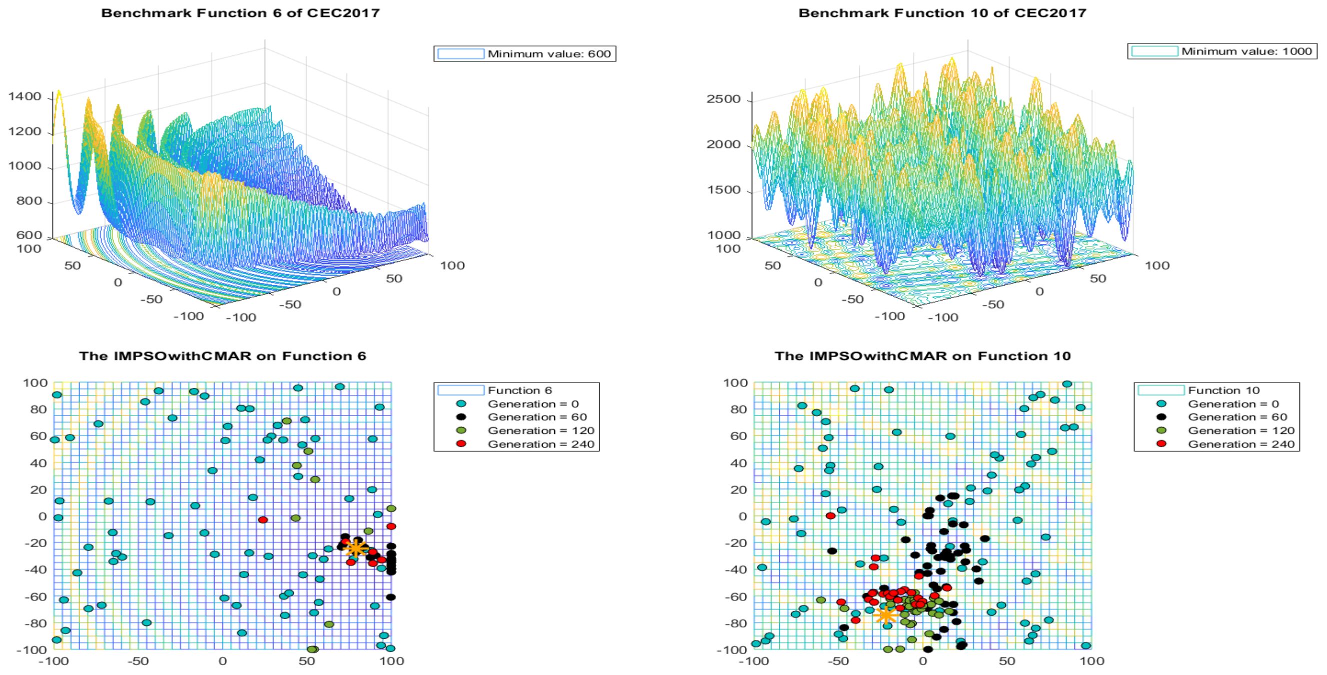

Figure 3: Plots of 2-dimensional particle evolution over various generations (0, 60, 120, 240) on benchmark f_6 and f_{10}.

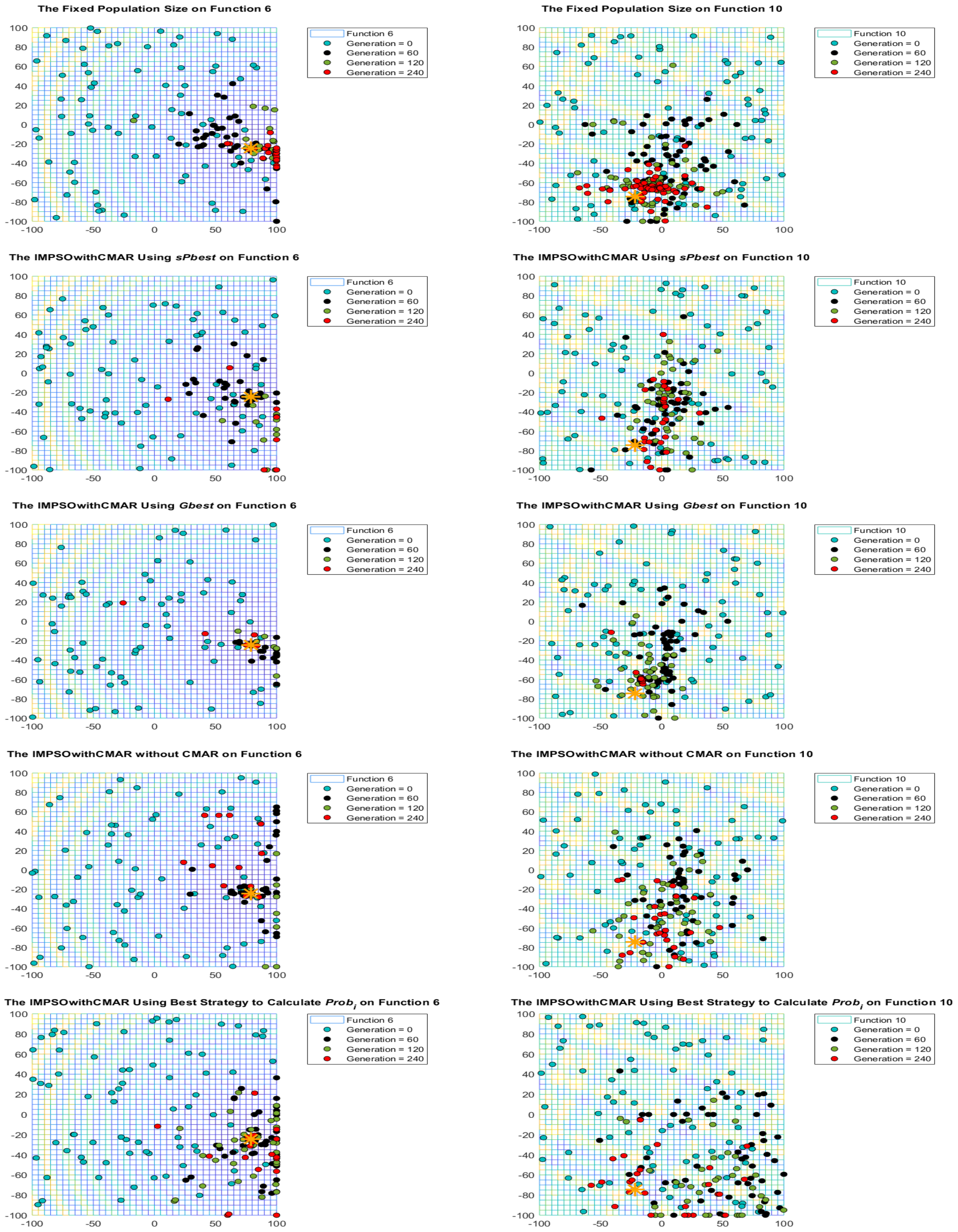

Figure 4: Plots of 2-dimensional particle evolution obtained using different mechanisms over various generations (0, 60, 120, 240) on benchmark f_6 and f_{10}.

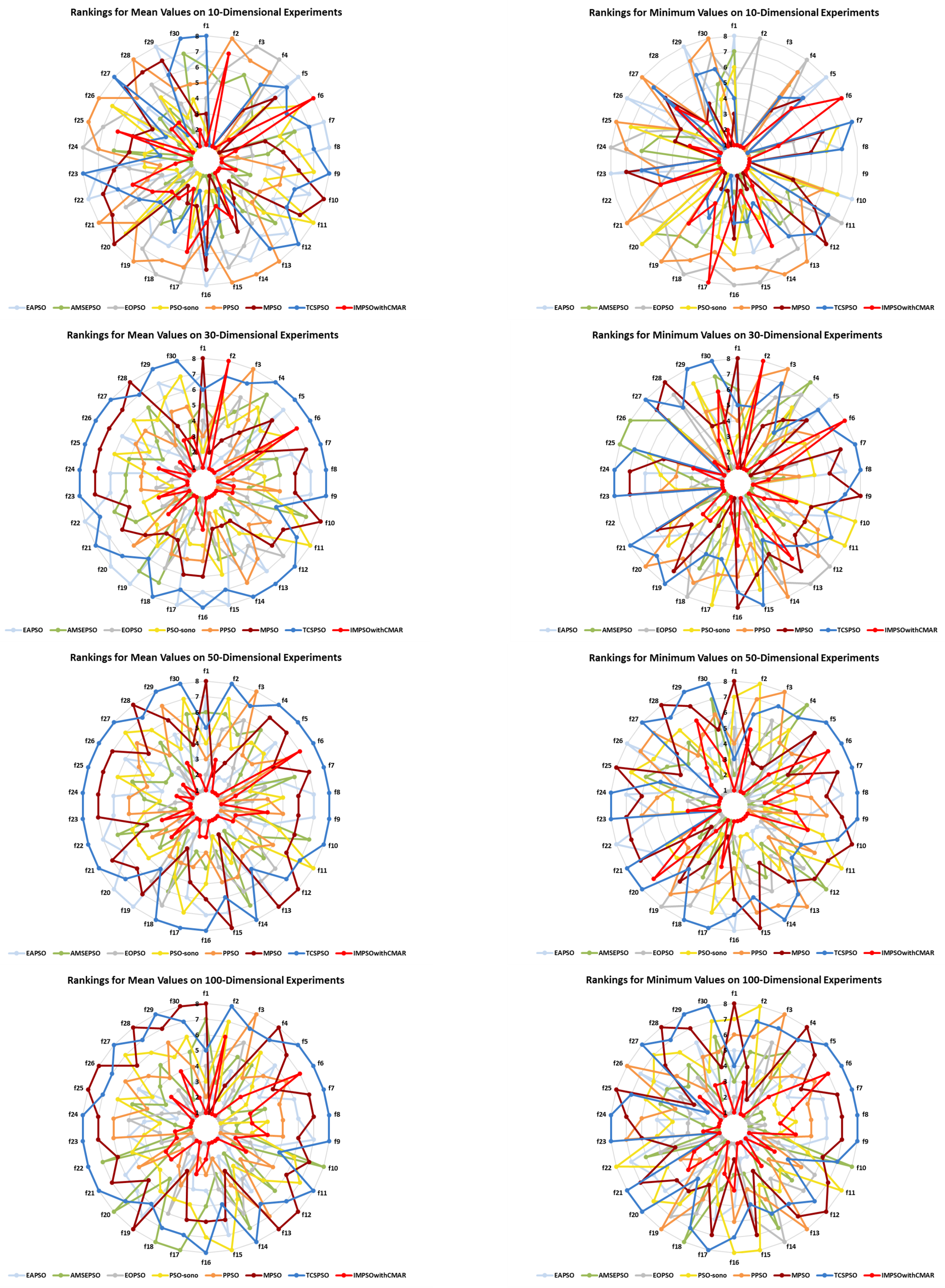

Figure 5: Radar map of the rankings for mean and minimum values on the 10-, 30-, 50-, and 100-dimensional experiments. The circumference and radius scale represent the benchmark functions and rankings, respectively.

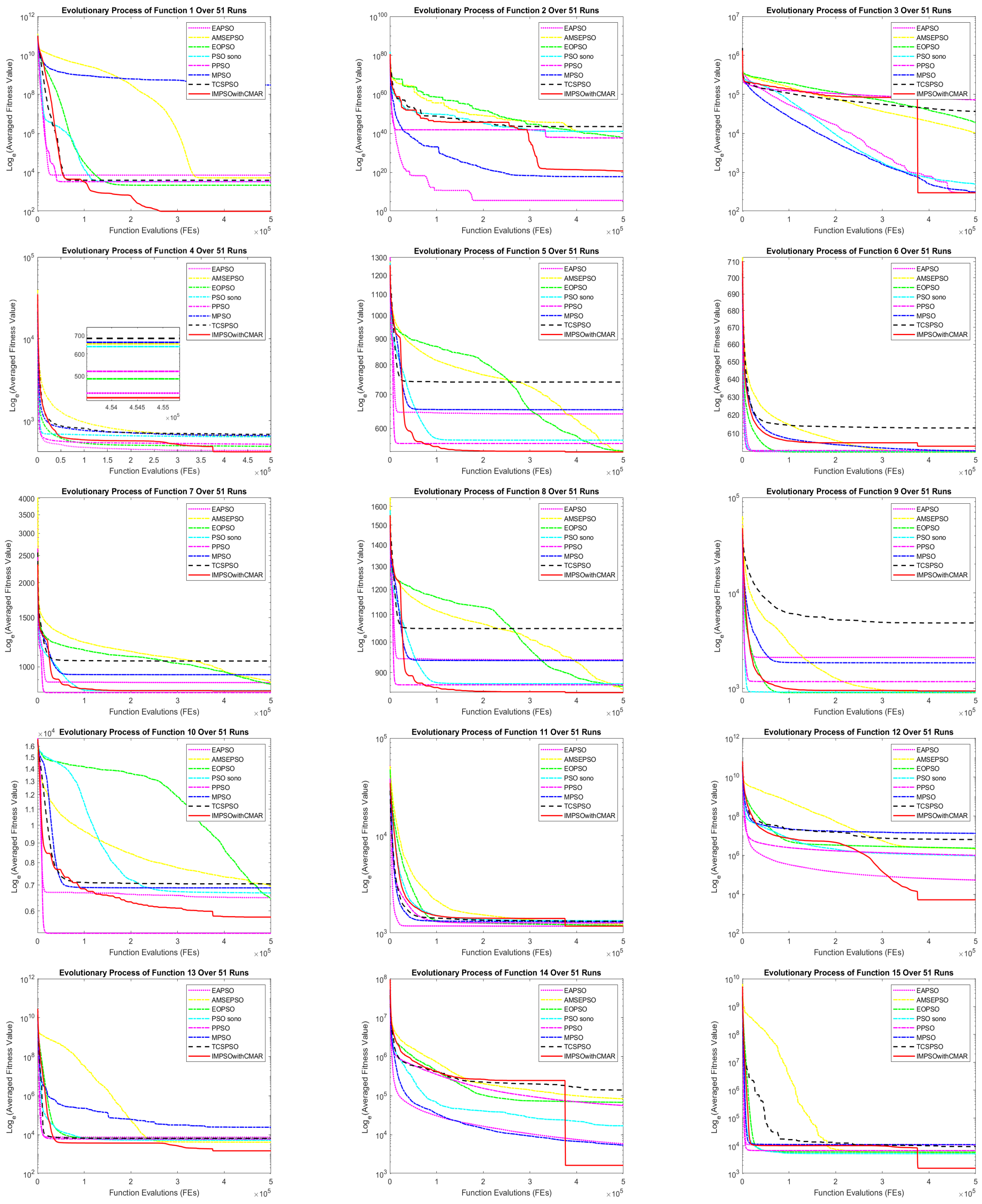

Figure 6: Evolutionary processes of 50-dimensional experiments, averaged over 51 runs, produced by the proposed IMPSOwithCMAR, EAPSO, AMSEPSO, EOPSO, PSO-sono, PPSO, MPSO, and TCSPSO for $f_1 - f_{15}$ on CEC2017 benchmark functions. The Y- and X-axes show the fitness values of the PSO-based variants and the function evaluations (FEs), respectively.

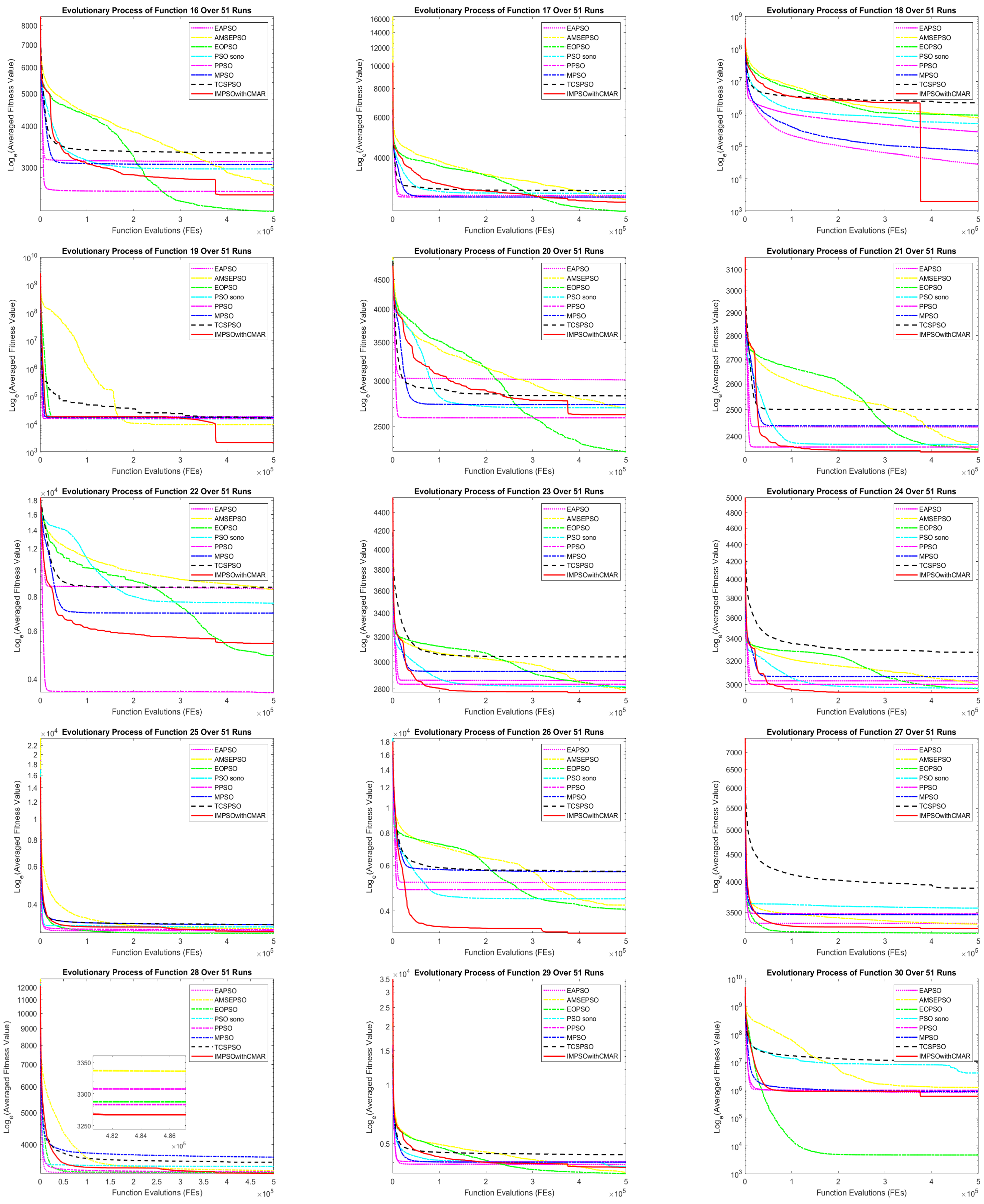

Figure 7: Evolutionary processes of 50-dimensional experiments, averaged over 51 runs, produced by the proposed IMPSOwithCMAR, EAPSO, AMSEPSO, EOPSO, PSO-sono, PPSO, MPSO, and TCSPSO for $f_{16} - f_{30}$ on CEC2017 benchmark functions. The Y- and X-axes show the fitness values of the PSO-based variants and the function evaluations (FEs), respectively.

significantly sudden and sharp convergence in most cases because of the intervention of the SQP method after 75% of the overall evolutionary process. Of the PSO-based variants, IMPSOwithC-MAR can best maintain the convergence rate over the course of evolution; this contrasts with some variants that produce a sharp convergence rate in the very early generations and remain nearly flat over the remainder of the evolution.

4.5. Computational Complexity

The computational complexity of the proposed IMPSOwithCMAR algorithm was determined by following the CEC2017 benchmark competition process proposed by [14]. All experiments were conducted using the following system: (a) CPU: Intel Core i7-8750h @ 2.20 GHz; (b) RAM: 16GB; (c) OS: Windows 10; (d) Software: MATLAB 2020b.

Table 6 presents the computational complexities of the IMPSOwithCMAR algorithm for 10, 30, 50, and 100 dimensions, respectively. In this table, T_0 denotes the time calculated using the following statements:

```
$x = 0.55;$
$for\ i = 1 : 1000000$
$x = x + x;\ x = x/2;\ x = x \times x;$
$x = sqrt(x);\ x = log(x);\ x = exp(x);\ x = x/(x+2);$
$end$
```

Table 6: Computational complexity for IMPSOwithCMAR.

Dimension	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
$10D$	0.0142	0.8005	1.9006	77.4718
$30D$	0.0142	1.2772	4.6574	238.0423
$50D$	0.0142	2.0070	7.0685	356.4437
$100D$	0.0142	5.9894	20.0237	988.3310

T_1 is the single run time of the benchmark function f_{18} with 2×10^5 evaluations in D dimensions. T_2 is the execution time of the IMPSOwithCMAR algorithm on function f_{18} for 2×10^5 evaluations with D dimensions. Time $(\hat{T}_2)$ is the average value of T_2 over five runs. T_0, T_1, $\hat{T}_2$, and $((\hat{T}_2 - T_1)/T_0)$, with their corresponding dimensions, are listed in Table 6. Comparisons of the computational complexities of the IMPSOwithCMAR algorithm and PSO-based variants for 30 dimensions are given in Table 7. The calculations of computational complexities for all of the PSO-based variants were based on the same software and hardware environment.

Table 7: Computational complexities of IMPSOwithCMAR, EAPSO, AMSEPSO, EOPSO, PSO-sono, PPSO, MPSO, and TCSPSO on 30 dimensions.

Algorithm	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
IMPSOwithCMAR	0.0142	1.2772	4.6574	238.0423
EAPSO	0.0142	1.2772	13.4683	858.5282
AMSEPSO	0.0142	1.2772	4.7851	247.0352
EOPSO	0.0142	1.2772	6.8389	391.6690
PSO-sono	0.0142	1.2772	2.0852	56.9014
PPSO	0.0142	1.2772	1.9830	49.7042
MPSO	0.0142	1.2772	1.7913	36.2042
TCSPSO	0.0142	1.2772	1.6935	29.3169

In Table 7, the computational complexity of IMPSOwithCMAR ranks 5^{th}, worse than TCSPSO, MPSO, PPSO, and PSO-sono but better than EOPSO, AMSEPSO, and EAPSO. Considering the excellent performance of the proposed algorithm, the computational cost is worthwhile and IMPSOwithCMAR remains competitive among the compared PSO-based variants.

5. Conclusions

This study devised a sophisticated general framework that integrates IMPSO, CMAR, and SQP methods. The framework essentially uses a global algorithm for exploration and a local search algorithm for exploitation periodically during evolution and further enhances the exploitation capability by applying the local search method in later generations of evolution. The effectiveness of the proposed mechanisms was validated experimentally against the most recent PSO-based variants on CEC2017 benchmark functions with various dimensions, with the results indicating that the combination of mechanisms used by the proposed algorithm is highly effective. The outstanding performance of the proposed ensemble strategy indicates that it can effectively coordinate searching. Future research directions for IMPSOwithCMAR include optimizing its exploitation capability by incorporating other local search methods, dynamically adjusting the timing of intervention rather than using a fixed rate of 75%, exploring additional combinations of effective mechanisms and complementary algorithms, and applying it to real-world problems.

Table 8: Experimental results of PSO-based variants for 10-dimensional CEC2017 benchmark functions based on 51 runs.

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Mean	**8.030425E+00**	3.186003E+03	1.590214E+03	8.842426E+02	6.667559E+02	9.341786E+02	7.650174E+02	4.110995E+03
	Rank|WRST	**1**	7 | >	6 | >	4 | >	2 | >	5 | >	3 | >	8 | >
	Min	**0.000000E+00**	7.062210E+00	9.620100E-01	1.105700E-01	1.915600E-01	6.440000E-03	7.846000E-02	9.731000E-02
f_1	Rank	**1**	8	7	5	6	2	3	4
	Median	**3.600000E-04**	2.074707E+03	6.299713E+02	5.664615E+02	3.946247E+02	3.324519E+02	2.300903E+02	1.461869E+03
	Rank	**1**	8	6	5	4	3	2	7
	(Std Dev)	(2.952925E+01)	(3.081125E+03)	(1.854472E+03)	(8.446592E+02)	(7.768769E+02)	(1.510431E+03)	(1.055219E+03)	(1.480714E+04)
	Mean	1.078235E+02	**0.000000E+00**	2.850463E+00	5.596703E+00	**0.000000E+00**	5.294706E+02	**0.000000E+00**	**0.000000E+00**
	Rank|WRST	7	**1** | <	5 | =	6 | >	**1** | <	8 | >	**1** | <	**1** | <
	Min	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	1.000000E-05	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
f_2	Rank	**1**	**1**	**1**	8	**1**	**1**	**1**	**1**
	Median	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	1.700000E-04	**0.000000E+00**	2.800000E+01	**0.000000E+00**	**0.000000E+00**
	Rank	**1**	**1**	**1**	7	**1**	8	**1**	**1**
	(Std Dev)	(5.050293E+02)	(0.000000E+00)	(1.642270E+01)	(1.633064E+01)	(0.000000E+00)	(1.085583E+03)	(0.000000E+00)	(0.000000E+00)
	Mean	**0.000000E+00**	**0.000000E+00**	1.960784E-07	1.058824E-05	**0.000000E+00**	2.352941E-06	**0.000000E+00**	**0.000000E+00**
	Rank|WRST	**1**	**1** | =	6 | =	8 | >	**1** | =	7 | =	**1** | =	**1** | =
	Min	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
f_3	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Median	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	1.000000E-05	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
	Rank	**1**	**1**	**1**	8	**1**	**1**	**1**	**1**
	(Std Dev)	(0.000000E+00)	(0.000000E+00)	(1.400280E-06)	(1.912190E-05)	(0.000000E+00)	(1.680336E-05)	(0.000000E+00)	(0.000000E+00)
	Mean	**0.000000E+00**	1.545098E-04	2.222725E+00	4.635064E+00	**0.000000E+00**	4.556058E+00	9.795653E-01	4.111587E+00
	Rank|WRST	**1**	3 | >	5 | >	8 | >	**1** | =	7 | >	4 | >	6 | >
	Min	**0.000000E+00**	1.000000E-05	2.390300E-01	4.080240E+00	**0.000000E+00**	1.403520E+00	1.159000E-02	3.409000E-02
f_4	Rank	**1**	3	6	8	**1**	7	4	5
	Median	**0.000000E+00**	1.000000E-05	1.855670E+00	4.659780E+00	**0.000000E+00**	4.683930E+00	1.050480E+00	3.276610E+00
	Rank	**1**	3	5	7	**1**	8	4	6
	(Std Dev)	(0.000000E+00)	(9.322260E-04)	(1.872530E+00)	(1.939482E-01)	(0.000000E+00)	(6.745759E-01)	(5.247337E-01)	(9.824638E+00)
	Mean	3.375059E+00	1.356526E+01	6.021541E+00	3.947365E+00	6.867235E+00	**2.750771E+00**	1.177328E+01	1.265811E+01
	Rank|WRST	2	8 | >	4 | =	3 | =	5 | >	**1** | <	6 | >	7 | >
	Min	9.949600E-01	4.974790E+00	**0.000000E+00**	9.780000E-03	**0.000000E+00**	**0.000000E+00**	3.979840E+00	3.979840E+00
f_5	Rank	5	8	**1**	4	**1**	**1**	6	6
	Median	2.984880E+00	1.130224E+01	2.984880E+00	2.991540E+00	5.969750E+00	**1.989920E+00**	1.094454E+01	1.193950E+01
	Rank	2	7	2	4	5	**1**	6	8
	(Std Dev)	(1.503388E+00)	(6.480218E+00)	(5.584247E+00)	(3.056227E+00)	(3.561108E+00)	(1.561651E+00)	(6.219368E+00)	(6.319550E+00)
	Mean	1.333224E-01	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	9.681961E-03	**0.000000E+00**	**0.000000E+00**	1.362745E-04
	Rank|WRST	8	**1** | <	**1** | <	**1** | <	7 | <	**1** | <	**1** | <	6 | <
	Min	4.300000E-02	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
f_6	Rank	8	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Median	1.295400E-01	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	8.000000E-05	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
	Rank	8	**1**	**1**	**1**	7	**1**	**1**	**1**
	(Std Dev)	(4.860957E-02)	(0.000000E+00)	(0.000000E+00)	(0.000000E+00)	(5.398522E-02)	(0.000000E+00)	(0.000000E+00)	(5.534400E-04)
	Mean	1.333679E+01	2.380771E+01	1.929721E+01	1.665715E+01	1.898554E+01	**1.237585E+01**	1.882434E+01	2.220984E+01
	Rank|WRST	2	8 | >	6 | >	3 | >	5 | >	**1** | <	4 | >	7 | >
	Min	1.088926E+01	1.119884E+01	1.081542E+01	1.096107E+01	1.247934E+01	**1.072588E+01**	1.181861E+01	1.251869E+01
f_7	Rank	3	5	2	4	7	**1**	6	8
	Median	1.304835E+01	2.510234E+01	1.707852E+01	1.544263E+01	1.871872E+01	**1.215899E+01**	1.742908E+01	2.270658E+01
	Rank	2	8	4	3	6	**1**	5	7
	(Std Dev)	(1.129400E+00)	(5.790018E+00)	(7.589243E+00)	(3.951069E+00)	(4.284811E+00)	(1.065566E+00)	(4.932944E+00)	(5.220098E+00)
	Mean	**2.614208E+00**	1.516487E+01	5.518895E+00	2.829198E+00	8.232794E+00	2.809298E+00	7.901143E+00	1.129247E+01
	Rank|WRST	**1**	8 | >	4 | >	3 | =	6 | >	2 | =	5 | >	7 | >
	Min	**0.000000E+00**	3.979840E+00	**0.000000E+00**	1.000000E-05	2.984880E+00	**0.000000E+00**	1.989920E+00	3.979840E+00
f_8	Rank	**1**	7	**1**	4	6	**1**	5	7
	Median	2.984880E+00	1.193950E+01	3.979840E+00	**2.151590E+00**	7.959670E+00	2.984880E+00	6.964710E+00	9.951730E+00
	Rank	2	8	4	**1**	6	2	5	7
	(Std Dev)	(1.378093E+00)	(7.952281E+00)	(4.326310E+00)	(2.145485E+00)	(3.781248E+00)	(1.588272E+00)	(3.767260E+00)	(4.600138E+00)
	Mean	**0.000000E+00**	1.755490E-03	**0.000000E+00**	**0.000000E+00**	6.560706E-02	3.510980E-03	1.241922E-02	1.508300E-01
	Rank|WRST	**1**	4 | =	**1** | =	**1** | =	7 | >	5 | =	6 | =	8 | >
	Min	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
f_9	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Median	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	(Std Dev)	(0.000000E+00)	(1.253671E-02)	(0.000000E+00)	(0.000000E+00)	(1.611353E-01)	(1.755139E-02)	(6.550906E-02)	(3.956263E-01)
	Mean	3.382576E+02	4.565924E+02	3.475630E+02	**2.357675E+02**	3.731008E+02	4.171844E+02	4.663117E+02	4.348828E+02
	Rank|WRST	2	7 | >	3 | =	**1** | <	4 | =	5 | =	8 | >	6 | >
	Min	**1.874000E-01**	1.043220E+01	3.216700E+00	**1.874000E-01**	7.142200E+00	3.539900E+00	3.747000E-01	**1.874000E-01**
f_{10}	Rank	**1**	8	5	**1**	7	6	4	**1**
	Median	3.403850E+02	4.693906E+02	3.264845E+02	**1.553845E+02**	3.145788E+02	3.588878E+02	4.495238E+02	4.301170E+02
	Rank	4	8	3	**1**	2	5	7	6
	(Std Dev)	(2.604214E+02)	(2.145745E+02)	(2.540047E+02)	(2.355809E+02)	(2.343548E+02)	(2.688872E+02)	(2.897015E+02)	(1.919531E+02)
	Mean	**7.249784E-01**	5.982271E+00	1.216608E+00	3.115469E+00	1.358213E+01	5.600198E+00	7.031447E+00	6.988449E+00
	Rank|WRST	**1**	5 | >	2 | >	3 | >	8 | >	4 | >	7 | >	6 | >
	Min	**0.000000E+00**	**0.000000E+00**	1.500000E-02	1.372500E+00	**0.000000E+00**	8.303000E-01	9.950000E-01	1.012900E+00
f_{11}	Rank	**1**	**1**	4	8	**1**	5	6	7
	Median	**9.950000E-01**	4.974800E+00	1.017300E+00	3.131700E+00	9.949600E+00	3.752200E+00	5.691300E+00	5.971000E+00

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Rank	**1**	5	2	3	8	4	6	7
	(Std Dev)	(9.589099E-01)	(4.900635E+00)	(1.218920E+00)	(8.239369E-01)	(1.354814E+01)	(6.115581E+00)	(4.718787E+00)	(4.973862E+00)
f_{12}	Mean	**3.857950E+02**	1.917425E+04	1.674944E+04	7.107753E+03	8.864540E+03	8.139338E+03	6.332340E+03	4.484475E+04
	Rank|WRST	**1**	7 | >	6 | >	3 | >	5 | >	4 | >	2 | >	8 | >
	Min	**9.710000E-02**	3.711147E+02	7.561086E+02	5.305249E+02	6.423936E+02	7.583048E+02	9.776507E+02	8.641670E+02
	Rank	**1**	2	5	3	4	6	8	7
	Median	**3.787012E+02**	1.443031E+04	1.227557E+04	5.173585E+03	4.304037E+03	6.421615E+03	5.316088E+03	8.692734E+03
	Rank	**1**	8	7	3	2	5	4	6
	(Std Dev)	(2.356323E+02)	(1.636531E+04)	(1.288290E+04)	(5.071686E+03)	(1.400404E+04)	(5.831981E+03)	(4.194948E+03)	(2.309930E+05)
f_{13}	Mean	**2.662986E+01**	4.128032E+03	2.664321E+03	3.357327E+03	1.454926E+02	7.059797E+03	9.714906E+02	5.743487E+03
	Rank|WRST	**1**	6 | >	4 | >	5 | >	2 | >	8 | >	3 | >	7 | >
	Min	3.016500E+00	8.540100E+00	9.832700E+00	4.846650E+01	8.387100E+00	1.116216E+03	**4.548000E-01**	1.245950E+01
	Rank	2	4	5	7	3	8	**1**	6
	Median	**1.849160E+01**	2.098068E+03	1.643375E+03	2.641808E+03	6.059060E+01	6.787603E+03	5.827290E+01	3.765186E+03
	Rank	**1**	5	4	6	3	8	2	7
	(Std Dev)	(3.540006E+01)	(4.740282E+03)	(3.011166E+03)	(3.106215E+03)	(2.083707E+02)	(3.819728E+03)	(2.710283E+03)	(5.996762E+03)
f_{14}	Mean	5.165260E+01	3.252531E+02	**3.031117E+01**	6.003068E+01	4.523177E+01	2.590017E+03	5.301457E+01	3.660970E+01
	Rank|WRST	4	7 | >	**1** | <	6 | =	3 | =	8 | >	5 | =	2 | <
	Min	2.216690E+01	5.276700E+00	**1.180200E+00**	2.956810E+01	5.270000E+00	1.313901E+02	1.488700E+00	3.281400E+00
	Rank	6	5	**1**	7	4	8	2	3
	Median	4.712380E+01	2.125068E+02	**2.937800E+01**	5.037640E+01	4.207000E+01	1.613935E+03	4.128690E+01	3.242390E+01
	Rank	5	7	**1**	6	4	8	3	2
	(Std Dev)	(2.128102E+01)	(2.915168E+02)	(1.572029E+01)	(3.016277E+01)	(1.915440E+01)	(2.357745E+03)	(3.937672E+01)	(2.531751E+01)
f_{15}	Mean	3.865389E+01	1.004205E+02	6.786804E+01	1.023983E+02	3.337829E+01	2.790128E+03	**2.821004E+01**	6.077042E+01
	Rank|WRST	3	6 | >	5 | =	7 | >	2 | =	8 | >	**1** | <	4 | >
	Min	1.491400E+00	3.013500E+00	2.582500E+00	1.812580E+01	2.130300E+00	1.414130E+01	**7.280000E-02**	2.257600E+00
	Rank	2	6	5	8	3	7	**1**	4
	Median	2.588970E+01	6.026520E+01	2.699460E+01	7.603030E+01	2.924860E+01	2.423773E+03	**1.109420E+01**	4.500070E+01
	Rank	2	6	3	7	4	8	**1**	5
	(Std Dev)	(3.246405E+01)	(1.156366E+02)	(1.312901E+02)	(1.048387E+02)	(2.984686E+01)	(2.250686E+03)	(3.446249E+01)	(5.108967E+01)
f_{16}	Mean	1.198463E+01	1.534177E+02	**1.570694E+00**	5.828927E+00	5.140461E+00	2.771460E+01	1.145095E+02	7.808951E+01
	Rank|WRST	4	8 | >	**1** | <	3 | =	2 | =	5 | =	7 | >	6 | =
	Min	2.786000E-01	**3.710000E-02**	1.322000E-01	1.231100E+00	3.612000E-01	6.334000E-01	3.267000E-01	2.907000E-01
	Rank	3	**1**	2	8	6	7	5	4
	Median	3.143600E+00	1.305157E+02	**1.118300E+00**	2.589200E+00	2.472400E+00	1.879500E+00	1.191674E+02	5.456400E+00
	Rank	5	8	**1**	4	3	2	7	6
	(Std Dev)	(1.999790E+01)	(1.323380E+02)	(1.317748E+00)	(1.782408E+01)	(1.724772E+01)	(6.060099E+01)	(1.104406E+02)	(9.187974E+01)
f_{17}	Mean	3.262066E+01	3.006345E+01	**2.101457E+01**	3.357767E+01	3.132780E+01	3.265439E+01	2.657675E+01	2.552018E+01
	Rank|WRST	6	4 | =	**1** | <	8 | =	5 | =	7 | =	3 | <	2 | <
	Min	7.183500E+00	1.356500E+00	1.326900E+00	2.886700E+00	2.400700E+00	2.478000E+00	6.441000E-01	**0.000000E+00**
	Rank	8	4	3	7	5	6	2	**1**
	Median	3.246210E+01	2.278400E+01	2.214800E+01	3.220160E+01	2.862380E+01	3.440920E+01	**1.978200E+01**	2.026490E+01
	Rank	7	4	3	6	5	8	**1**	2
	(Std Dev)	(1.037734E+01)	(2.058865E+01)	(8.749788E+00)	(9.931359E+00)	(1.271983E+01)	(1.740250E+01)	(2.261908E+01)	(2.101450E+01)
f_{18}	Mean	7.317874E+01	4.756437E+03	2.583152E+03	2.272704E+04	**6.182331E+01**	8.633783E+03	3.408772E+02	4.497697E+03
	Rank|WRST	2	6 | >	4 | >	8 | >	**1** | =	7 | >	3 | >	5 | >
	Min	2.392460E+01	2.544140E+01	1.310988E+02	8.257066E+02	**1.701700E+00**	6.302566E+02	3.360700E+00	2.466040E+01
	Rank	3	5	6	8	**1**	7	2	4
	Median	5.404100E+01	2.584950E+03	2.165398E+03	2.361712E+04	**4.059000E+01**	6.660525E+03	7.530420E+01	2.560113E+03
	Rank	2	6	4	8	**1**	7	3	5
	(Std Dev)	(6.040278E+01)	(6.945118E+03)	(2.290436E+03)	(1.360979E+04)	(5.519578E+01)	(6.422606E+03)	(7.030645E+02)	(5.505540E+03)
f_{19}	Mean	3.915514E+01	5.750064E+02	9.865004E+01	2.803438E+03	**2.076878E+01**	4.423817E+03	3.275042E+01	7.571447E+01
	Rank|WRST	3	6 | >	5 | >	7 | >	**1** | <	8 | >	2 | <	4 | =
	Min	8.398000E+00	6.692300E+00	8.941700E+00	2.192620E+01	2.919400E+00	2.909042E+02	**6.081000E-01**	3.879100E+00
	Rank	5	4	6	7	2	8	**1**	3
	Median	3.171500E+01	2.225288E+02	4.069310E+01	2.677830E+02	**1.151360E+01**	3.649135E+03	1.349460E+01	3.630740E+01
	Rank	3	6	5	7	**1**	8	2	4
	(Std Dev)	(3.256727E+01)	(7.714016E+02)	(1.468097E+02)	(4.382235E+03)	(1.710200E+01)	(3.420357E+03)	(3.640678E+01)	(8.429534E+01)
f_{20}	Mean	1.053690E+01	1.592043E+01	1.053051E+01	**4.789157E+00**	2.385850E+01	1.870505E+01	4.926970E+01	1.195030E+01
	Rank|WRST	3	5 | >	2 | >	**1** | >	7 | >	6 | >	8 | >	4 | >
	Min	**0.000000E+00**	**0.000000E+00**	3.122000E-01	1.000000E-04	9.950000E-01	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
	Rank	**1**	**1**	7	6	8	**1**	**1**	**1**
	Median	**3.122000E-01**	1.305440E+01	4.749500E+00	1.307200E+00	2.289800E+01	2.130660E+01	2.199070E+01	9.804100E+00
	Rank	**1**	5	3	2	8	6	7	4
	(Std Dev)	(2.741999E+01)	(1.784236E+01)	(1.022509E+01)	(6.913143E+00)	(8.534039E+00)	(1.311336E+01)	(5.487816E+01)	(1.078298E+01)
f_{21}	Mean	1.854903E+02	**1.115001E+02**	1.807796E+02	1.958892E+02	1.788440E+02	2.033473E+02	2.009881E+02	1.875734E+02
	Rank|WRST	4	**1** | <	3 | >	6 | =	2 | >	8 | >	7 | >	5 | >
	Min	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	1.202205E+02	**1.000000E+02**	1.662160E+02	**1.000000E+02**	**1.000000E+02**
	Rank	**1**	**1**	**1**	7	**1**	8	**1**	**1**
	Median	2.037700E+02	**1.000000E+02**	2.055682E+02	2.034714E+02	2.067434E+02	2.046612E+02	2.113995E+02	2.136447E+02
	Rank	3	**1**	5	2	6	4	7	8
	(Std Dev)	(3.997780E+01)	(3.253518E+01)	(4.669400E+01)	(2.378970E+01)	(4.857588E+01)	(7.384543E+00)	(3.276100E+01)	(5.134175E+01)
f_{22}	Mean	1.000834E+02	1.102969E+02	**9.035321E+01**	1.000175E+02	9.724182E+01	1.000238E+02	1.017320E+02	1.008972E+02
	Rank|WRST	5	8 | >	**1** | =	3 | =	2 | =	4 | =	7 | >	6 | >
	Min	1.000000E+02	1.609490E+01	**0.000000E+00**	1.000000E+02	2.344390E+01	1.000000E+02	1.000000E+02	4.403180E+01
	Rank	5	2	**1**	5	3	5	5	4

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Median	**1.000000E+02**	1.010939E+02	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	1.014904E+02	1.018331E+02
	Rank	**1**	6	**1**	**1**	**1**	**1**	7	8
	(Std Dev)	(1.802562E-01)	(7.803591E+01)	(2.560306E+01)	(9.036933E-02)	(1.462887E+01)	(8.265572E-02)	(1.168767E+00)	(8.183718E+00)
f_{23}	Mean	3.055547E+02	3.178994E+02	**3.053426E+02**	3.082509E+02	3.077281E+02	3.074187E+02	3.168224E+02	3.191282E+02
	Rank|WRST	2	7 | >	**1** | =	5 | >	4 | >	3 | >	6 | >	8 | >
	Min	**3.000000E+02**	3.061013E+02	**3.000000E+02**	3.005675E+02	**3.000000E+02**	3.028820E+02	3.032925E+02	3.031438E+02
	Rank	**1**	8	**1**	4	**1**	5	7	6
	Median	3.055044E+02	3.161386E+02	**3.042866E+02**	3.078400E+02	3.073628E+02	3.066188E+02	3.159965E+02	3.196743E+02
	Rank	2	7	**1**	5	4	3	6	8
	(Std Dev)	(3.123879E+00)	(8.617302E+00)	(4.363889E+00)	(2.913365E+00)	(3.535068E+00)	(3.099960E+00)	(7.607058E+00)	(9.128830E+00)
f_{24}	Mean	**2.112811E+02**	2.932541E+02	3.317451E+02	3.366359E+02	3.171194E+02	3.333928E+02	3.200797E+02	3.165898E+02
	Rank|WRST	**1**	2 | >	6 | >	8 | >	4 | >	7 | >	5 | >	3 | >
	Min	**1.000000E+02**	**1.000000E+02**	1.437338E+02	3.309445E+02	**1.000000E+02**	2.671716E+02	**1.000000E+02**	**1.000000E+02**
	Rank	**1**	**1**	6	8	**1**	7	**1**	**1**
	Median	**2.004832E+02**	3.416905E+02	3.354788E+02	3.359612E+02	3.345330E+02	3.352405E+02	3.393917E+02	3.471694E+02
	Rank	**1**	7	4	5	2	3	6	8
	(Std Dev)	(1.074877E+02)	(1.024942E+02)	(3.117424E+01)	(3.316385E+00)	(6.406715E+01)	(1.105945E+01)	(7.009633E+01)	(8.787271E+01)
f_{25}	Mean	4.307372E+02	4.231977E+02	**4.161502E+02**	4.386374E+02	4.288044E+02	4.427171E+02	4.306370E+02	4.298644E+02
	Rank|WRST	6	2 | =	**1** | <	7 | >	3 | >	8 | >	5 | >	4 | >
	Min	3.978048E+02	**3.977429E+02**	3.979891E+02	3.981162E+02	3.982605E+02	3.995850E+02	3.978862E+02	**3.977429E+02**
	Rank	3	**1**	5	6	7	8	4	**1**
	Median	4.434604E+02	4.434163E+02	**3.982840E+02**	4.461263E+02	4.458687E+02	4.463809E+02	4.438512E+02	4.438965E+02
	Rank	3	2	**1**	7	6	8	4	5
	(Std Dev)	(2.059103E+01)	(2.369270E+01)	(2.301833E+01)	(1.697256E+01)	(2.366233E+01)	(1.275915E+01)	(2.149849E+01)	(2.238849E+01)
f_{26}	Mean	**2.783119E+02**	3.257112E+02	3.018052E+02	3.460452E+02	3.758342E+02	3.840358E+02	3.172017E+02	3.066712E+02
	Rank|WRST	**1**	5 | >	2 | =	6 | >	7 | >	8 | >	4 | >	3 | >
	Min	**0.000000E+00**	3.000000E+02	1.256853E+02	2.000000E+02	2.000000E+02	2.000000E+02	2.000000E+02	**0.000000E+00**
	Rank	**1**	8	3	4	4	4	4	**1**
	Median	**3.000000E+02**	**3.000000E+02**	**3.000000E+02**	**3.000000E+02**	**3.000000E+02**	3.554061E+02	**3.000000E+02**	**3.000000E+02**
	Rank	**1**	**1**	**1**	**1**	**1**	8	**1**	**1**
	(Std Dev)	(6.172657E+01)	(1.401935E+02)	(3.235885E+01)	(1.788978E+02)	(2.568168E+02)	(1.494805E+02)	(7.240163E+01)	(6.794476E+01)
f_{27}	Mean	3.925314E+02	3.976512E+02	3.936206E+02	3.911447E+02	**3.816387E+02**	4.003677E+02	4.033279E+02	4.101003E+02
	Rank|WRST	3	5 | =	4 | >	2 | =	**1** | <	6 | >	7 | >	8 | >
	Min	3.890055E+02	3.889780E+02	3.881824E+02	**3.747407E+02**	3.773976E+02	3.930774E+02	3.899557E+02	3.912438E+02
	Rank	5	4	3	**1**	2	8	6	7
	Median	3.907519E+02	3.938188E+02	3.940641E+02	3.899549E+02	**3.806957E+02**	3.985931E+02	3.985701E+02	4.032594E+02
	Rank	3	4	5	2	**1**	7	6	8
	(Std Dev)	(4.889170E+00)	(1.941679E+01)	(2.069566E+00)	(4.467548E+00)	(3.717960E+00)	(5.375906E+00)	(1.722852E+01)	(2.101891E+01)
f_{28}	Mean	3.859214E+02	**3.644457E+02**	3.866099E+02	4.677591E+02	4.513302E+02	5.799543E+02	5.319541E+02	3.782205E+02
	Rank|WRST	3	**1** | =	4 | =	6 | >	5 | =	8 | >	7 | >	2 | =
	Min	3.000000E+02	3.000000E+02	3.000000E+02	3.000000E+02	**0.000000E+00**	3.000000E+02	3.000000E+02	**0.000000E+00**
	Rank	3	3	3	3	**1**	3	3	**1**
	Median	**3.000000E+02**	**3.000000E+02**	3.000009E+02	4.894649E+02	5.000025E+02	6.118218E+02	5.837453E+02	**3.000000E+02**
	Rank	**1**	**1**	4	5	6	8	7	**1**
	(Std Dev)	(1.216878E+02)	(1.142481E+02)	(1.236347E+02)	(6.951826E+01)	(1.514079E+02)	(8.595036E+01)	(1.189356E+02)	(1.553827E+02)
f_{29}	Mean	**2.537280E+02**	3.155738E+02	2.570077E+02	2.718614E+02	2.640478E+02	2.825964E+02	2.882800E+02	2.838342E+02
	Rank|WRST	**1**	8 | >	2 | =	4 | >	3 | >	5 | >	7 | >	6 | >
	Min	**2.301093E+02**	2.562077E+02	2.408820E+02	2.456028E+02	2.333766E+02	2.528293E+02	2.440378E+02	2.478829E+02
	Rank	**1**	8	3	5	2	7	4	6
	Median	**2.512300E+02**	3.024326E+02	2.558728E+02	2.726170E+02	2.625811E+02	2.766217E+02	2.816203E+02	2.823104E+02
	Rank	**1**	8	2	4	3	5	6	7
	(Std Dev)	(1.618427E+01)	(4.781662E+01)	(9.078366E+00)	(1.401677E+01)	(2.283939E+01)	(2.332891E+01)	(3.773787E+01)	(2.303928E+01)
f_{30}	Mean	4.129209E+04	2.194375E+05	4.187915E+05	**3.627572E+04**	1.031412E+05	1.468334E+05	8.105805E+04	5.114002E+05
	Rank|WRST	2	6 | >	7 | >	**1** | >	4 | >	5 | >	3 | =	8 | >
	Min	4.383606E+02	4.701413E+02	9.501850E+02	2.590445E+03	6.562228E+02	2.805441E+03	**4.112270E+02**	1.088077E+03
	Rank	2	3	5	7	4	8	**1**	6
	Median	7.987353E+02	1.921571E+03	1.046671E+04	9.127699E+03	5.994396E+03	7.600505E+03	**7.122393E+02**	1.976965E+04
	Rank	2	3	7	6	4	5	**1**	8
	(Std Dev)	(2.069299E+05)	(4.059323E+05)	(6.271299E+05)	(1.224610E+05)	(2.556025E+05)	(3.368551E+05)	(2.452575E+05)	(1.145770E+06)

Table 9: Experimental results for PSO-based variants on 30-dimensional CEC2017 benchmark functions based on 51 runs.

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
f_1	Mean	**0.000000E+00**	4.192638E+03	3.695293E+03	3.097175E+03	2.221864E+03	2.619961E+03	6.277538E+05	3.764354E+03
	Rank|WRST	**1**	7 | >	5 | >	4 | >	2 | >	3 | >	8 | >	6 | >
	Min	**0.000000E+00**	1.296785E+01	8.922230E+00	2.284700E-01	3.570000E-01	1.236140E+00	3.004118E+01	1.827640E+00
	Rank	**1**	7	6	2	3	4	8	5
	Median	**0.000000E+00**	3.007053E+03	2.903186E+03	1.443856E+03	9.852258E+02	1.264611E+03	8.828669E+03	2.486026E+03
	Rank	**1**	7	6	4	2	3	8	5
	(Std Dev)	(0.000000E+00)	(3.989163E+03)	(3.468143E+03)	(3.463057E+03)	(2.799437E+03)	(3.074762E+03)	(2.105396E+06)	(4.011028E+03)
f_2	Mean	1.817500E+19	**2.839216E+01**	1.354495E+15	8.652257E+14	4.612299E+16	1.808480E+16	1.275031E+04	3.122203E+18
	Rank|WRST	8	**1** | <	4 | <	3 | <	6 | <	5 | <	2 | <	7 | <
	Min	2.665445E+16	**0.000000E+00**	1.500000E-04	8.800000E-04	4.043700E+05	5.371664E+06	**0.000000E+00**	5.400000E+01
	Rank	8	**1**	3	4	6	7	**1**	5

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Median	5.159370E+18	**2.300000E+01**	1.228838E+12	1.206637E+04	6.250622E+12	7.765111E+12	5.200000E+01	7.143297E+11
	Rank	8	**1**	5	3	6	7	2	4
	(Std Dev)	(5.890754E+19)	(2.713085E+01)	(5.596140E+15)	(6.176568E+15)	(1.665560E+17)	(6.992940E+16)	(5.452242E+04)	(2.117114E+19)
	Mean	**0.000000E+00**	**0.000000E+00**	2.185905E+02	9.154339E+02	1.860373E-02	2.860689E+04	3.847059E-03	3.594462E+03
	Rank|WRST	**1**	**1** | =	5 | >	6 | >	4 | >	8 | >	3 | >	7 | >
	Min	**0.000000E+00**	**0.000000E+00**	1.900000E-04	1.340685E+02	**0.000000E+00**	9.674485E+03	1.000000E-05	9.934281E+02
f_3	Rank	**1**	**1**	5	6	**1**	8	4	7
	Median	**0.000000E+00**	**0.000000E+00**	4.765220E+01	6.122227E+02	2.460000E-03	2.518618E+04	7.300000E-04	3.363253E+03
	Rank	**1**	**1**	5	6	4	8	3	7
	(Std Dev)	(0.000000E+00)	(0.000000E+00)	(3.750906E+02)	(1.311164E+03)	(5.358193E-02)	(1.243633E+04)	(1.111432E-02)	(1.600039E+03)
	Mean	**5.164149E+00**	3.257132E+01	1.205902E+02	6.129346E+01	1.167869E+02	8.594702E+01	6.155475E+01	1.416250E+02
	Rank|WRST	**1**	2 | >	7 | >	3 | >	6 | >	5 | >	4 | >	8 | >
	Min	**0.000000E+00**	2.000000E-05	7.511644E+01	2.280390E+01	3.987530E+00	2.240000E-03	3.128320E+00	2.766100E-01
f_4	Rank	**1**	2	8	7	6	3	5	4
	Median	**3.986620E+00**	5.856156E+01	1.195741E+02	7.578031E+01	1.235340E+02	8.561104E+01	7.031353E+01	1.373633E+02
	Rank	**1**	2	6	4	7	5	3	8
	(Std Dev)	(1.334967E+01)	(2.901910E+01)	(1.491324E+01)	(3.462478E+01)	(4.428338E+01)	(2.657092E+01)	(2.510974E+01)	(5.862193E+01)
	Mean	**1.646559E+01**	6.982934E+01	3.226141E+01	2.471053E+01	3.341902E+01	2.292306E+01	6.204565E+01	9.605474E+01
	Rank|WRST	**1**	7 | >	4 | =	3 | =	5 | >	2 | >	6 | >	8 | >
	Min	6.964710E+00	3.681338E+01	**5.969750E+00**	8.214540E+00	1.193951E+01	1.392942E+01	1.790926E+01	3.382859E+01
f_5	Rank	2	8	**1**	3	4	5	6	7
	Median	1.691430E+01	6.516285E+01	1.591934E+01	**1.516653E+01**	3.084372E+01	2.387900E+01	5.671257E+01	9.352577E+01
	Rank	3	7	2	**1**	5	4	6	8
	(Std Dev)	(3.885760E+00)	(2.005085E+01)	(3.679353E+01)	(2.501768E+01)	(1.087952E+01)	(4.683539E+00)	(2.039658E+01)	(2.500387E+01)
	Mean	1.182968E+00	**0.000000E+00**	3.538235E-03	1.960784E-07	3.137708E-01	7.200020E-02	1.669098E-02	3.217085E+00
	Rank|WRST	7	**1** | <	3 | <	2 | <	6 | <	5 | <	4 | <	8 | >
	Min	5.997000E-01	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	2.000000E-05	4.000000E-05	**0.000000E+00**	1.788100E-01
f_6	Rank	8	**1**	**1**	**1**	5	6	**1**	7
	Median	1.142630E+00	**0.000000E+00**	2.800000E-04	**0.000000E+00**	2.753000E-02	3.570000E-03	**0.000000E+00**	2.401700E+00
	Rank	7	**1**	4	**1**	6	5	**1**	8
	(Std Dev)	(2.652778E-01)	(0.000000E+00)	(1.115508E-02)	(1.400280E-06)	(5.351655E-01)	(2.215908E-01)	(7.457982E-02)	(2.344268E+00)
	Mean	5.687600E+01	8.639119E+01	8.527437E+01	8.464876E+01	6.209841E+01	**4.934260E+01**	1.072620E+02	1.461693E+02
	Rank|WRST	2	6 | >	5 | =	4 | =	3 | >	**1** | <	7 | >	8 | >
	Min	4.272697E+01	5.804116E+01	**3.557507E+01**	4.297954E+01	4.642703E+01	3.916581E+01	6.317594E+01	8.349864E+01
f_7	Rank	3	6	**1**	4	5	2	7	8
	Median	5.622703E+01	8.366882E+01	5.133328E+01	5.498878E+01	5.969471E+01	**4.903654E+01**	1.075044E+02	1.413327E+02
	Rank	4	6	2	3	5	**1**	7	8
	(Std Dev)	(6.375188E+00)	(1.918300E+01)	(5.076902E+01)	(4.755588E+01)	(1.154293E+01)	(5.981583E+00)	(2.267898E+01)	(3.780321E+01)
	Mean	**1.623149E+01**	7.464263E+01	4.241190E+01	2.867612E+01	3.417059E+01	2.335226E+01	5.812591E+01	9.161653E+01
	Rank|WRST	**1**	7 | >	5 | >	3 | =	4 | >	2 | >	6 | >	8 | >
	Min	6.964710E+00	3.681343E+01	8.954630E+00	**6.884640E+00**	1.591934E+01	1.293447E+01	2.487394E+01	5.074283E+01
f_8	Rank	2	7	3	**1**	5	4	6	8
	Median	1.591934E+01	7.163690E+01	1.890422E+01	**1.543250E+01**	3.283362E+01	2.288405E+01	5.969743E+01	8.855096E+01
	Rank	2	7	3	**1**	5	4	6	8
	(Std Dev)	(4.195328E+00)	(1.836206E+01)	(4.015422E+01)	(2.807989E+01)	(1.052820E+01)	(5.934328E+00)	(1.519283E+01)	(2.026984E+01)
	Mean	4.072802E-01	1.507273E+02	8.146845E-01	**4.089941E-02**	2.913693E+00	8.720690E+00	1.261733E+02	5.159087E+02
	Rank|WRST	2	7 | >	3 | =	**1** | <	4 | >	5 | >	6 | >	8 | >
	Min	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	1.537515E+01	1.236111E+01
f_9	Rank	**1**	**1**	**1**	**1**	**1**	**1**	8	7
	Median	8.953000E-02	1.817270E+00	4.543200E-01	**0.000000E+00**	9.086500E-01	4.903790E+00	6.868297E+01	3.572465E+02
	Rank	2	5	3	**1**	4	6	7	8
	(Std Dev)	(1.014528E+00)	(6.078512E+02)	(1.212529E+00)	(1.236416E-01)	(5.285673E+00)	(9.664145E+00)	(1.248995E+02)	(4.539553E+02)
	Mean	2.476046E+03	3.061148E+03	3.442309E+03	2.995413E+03	3.194753E+03	**2.211122E+03**	3.444195E+03	3.096726E+03
	Rank|WRST	2	4 | >	7 | >	3 | =	6 | >	**1** | <	8 | >	5 | >
	Min	1.699657E+03	1.835100E+03	**8.095702E+02**	1.274986E+03	2.092612E+03	1.325678E+03	1.743933E+03	1.781491E+03
f_{10}	Rank	4	7	**1**	2	8	3	5	6
	Median	2.431745E+03	2.981240E+03	3.612530E+03	2.706002E+03	3.148064E+03	**2.103415E+03**	3.429177E+03	3.156079E+03
	Rank	2	4	8	3	5	**1**	7	6
	(Std Dev)	(3.747945E+02)	(7.083997E+02)	(7.951716E+02)	(1.221830E+03)	(6.523951E+02)	(5.414346E+02)	(7.185893E+02)	(6.565126E+02)
	Mean	**2.694394E+01**	5.191102E+01	3.841060E+01	6.251909E+01	1.572243E+02	8.514451E+01	9.774841E+01	1.135041E+02
	Rank|WRST	**1**	3 | >	2 | =	4 | >	8 | >	5 | >	6 | >	7 | >
	Min	5.969800E+00	9.949700E+00	**5.305400E+00**	8.689700E+00	8.059260E+01	3.399660E+01	2.411820E+01	3.429330E+01
f_{11}	Rank	2	4	**1**	3	8	6	5	7
	Median	1.989910E+01	4.477310E+01	**1.598160E+01**	7.163360E+01	1.472701E+02	8.385890E+01	9.514550E+01	1.043675E+02
	Rank	2	3	**1**	4	8	5	6	7
	(Std Dev)	(2.195786E+01)	(3.178818E+01)	(3.459324E+01)	(2.614970E+01)	(5.552407E+01)	(3.139318E+01)	(4.511406E+01)	(5.355727E+01)
	Mean	**1.256211E+03**	1.590648E+04	1.248784E+05	5.845956E+05	2.097481E+05	1.869790E+05	2.515126E+05	1.130193E+06
	Rank|WRST	**1**	2 | >	3 | >	7 | >	5 | >	4 | >	6 | >	8 | >
	Min	**3.725759E+02**	1.923140E+03	4.611429E+03	4.591834E+04	2.226055E+03	3.815855E+04	8.097608E+03	1.293985E+04
f_{12}	Rank	**1**	2	4	8	3	7	5	6
	Median	**1.300392E+03**	1.368764E+04	5.473694E+04	4.724911E+05	1.997952E+04	1.595247E+05	4.527329E+04	1.052106E+05
	Rank	**1**	2	5	8	3	7	4	6
	(Std Dev)	(6.109749E+02)	(8.905981E+03)	(1.716400E+05)	(4.120601E+05)	(6.837585E+05)	(1.198876E+05)	(6.271533E+05)	(2.933862E+06)
	Mean	**2.592484E+02**	2.261881E+04	8.684131E+03	1.540068E+04	1.259294E+04	1.643004E+04	1.207386E+04	3.326245E+04
	Rank|WRST	**1**	7 | >	2 | >	5 | >	4 | >	6 | >	3 | >	8 | >
f_{13}	Min	7.381720E+01	4.127610E+01	**3.048210E+01**	1.616023E+02	4.910690E+01	5.303350E+01	8.831280E+01	6.605290E+01

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Rank	6	2	**1**	8	3	4	7	5
	Median	**2.073308E+02**	1.197458E+04	8.307972E+03	1.108670E+04	5.502412E+03	1.649382E+04	8.470444E+03	7.765100E+03
	Rank	**1**	7	4	6	2	8	5	3
	(Std Dev)	(1.573997E+02)	(2.205546E+04)	(5.921467E+03)	(1.359208E+04)	(1.469097E+04)	(1.363834E+04)	(1.126138E+04)	(1.411195E+05)
	Mean	**1.475881E+02**	1.417558E+03	7.200764E+03	1.264486E+04	3.758482E+02	1.713629E+04	4.742664E+02	1.856169E+04
	Rank|WRST	**1**	4 | >	5 | >	6 | >	2 | =	7 | >	3 | >	8 | >
	Min	7.787920E+01	**4.074070E+01**	2.202025E+02	2.299556E+02	5.148230E+01	2.457233E+03	1.275583E+02	7.987980E+01
f_{14}	Rank	3	**1**	6	7	2	8	5	4
	Median	**1.433738E+02**	1.038012E+03	4.491816E+03	7.999592E+03	1.613884E+02	1.093366E+04	3.457284E+02	1.027973E+04
	Rank	**1**	4	5	6	2	8	3	7
	(Std Dev)	(4.264508E+01)	(1.376080E+03)	(8.420474E+03)	(1.667843E+04)	(1.464864E+03)	(1.677878E+04)	(3.446774E+02)	(2.254480E+04)
	Mean	**9.782894E+00**	1.341372E+04	2.882950E+03	2.738305E+03	5.691458E+03	1.366505E+03	1.941993E+03	5.798165E+03
	Rank|WRST	**1**	8 | >	5 | >	4 | >	6 | >	2 | >	3 | >	7 | >
	Min	**2.447200E+00**	7.343040E+01	1.284220E+01	3.830560E+01	1.000448E+02	5.706250E+01	8.615880E+01	1.090553E+02
f_{15}	Rank	**1**	5	2	3	7	4	6	8
	Median	**7.393400E+00**	7.249262E+03	1.731952E+03	1.811072E+03	3.818653E+03	4.924974E+02	1.110196E+03	3.071875E+03
	Rank	**1**	8	4	5	7	2	3	6
	(Std Dev)	(1.052521E+01)	(1.280620E+04)	(3.264408E+03)	(3.197992E+03)	(6.844508E+03)	(2.206197E+03)	(1.972500E+03)	(7.345047E+03)
	Mean	5.374801E+02	8.228238E+02	4.575426E+02	**3.923967E+02**	6.283391E+02	7.517736E+02	7.800699E+02	9.109083E+02
	Rank|WRST	3	7 | >	2 | =	**1** | <	4 | =	5 | >	6 | >	8 | >
	Min	2.384380E+01	1.359240E+02	1.744020E+01	**7.560000E+00**	2.175320E+01	1.361361E+02	3.456891E+02	2.939466E+02
f_{16}	Rank	4	5	2	**1**	3	6	8	7
	Median	5.073835E+02	8.838458E+02	4.306009E+02	**3.500509E+02**	6.076669E+02	6.879831E+02	7.562519E+02	9.235590E+02
	Rank	3	7	2	**1**	4	5	6	8
	(Std Dev)	(2.338863E+02)	(3.276928E+02)	(3.132752E+02)	(2.663151E+02)	(2.467308E+02)	(3.113308E+02)	(2.321222E+02)	(2.729017E+02)
	Mean	1.325463E+02	3.730581E+02	**1.100949E+02**	1.561763E+02	2.060534E+02	2.399524E+02	3.198271E+02	3.319945E+02
	Rank|WRST	2	8 | >	**1** | =	3 | =	4 | >	5 | >	6 | >	7 | >
	Min	3.522650E+01	4.908670E+01	3.733370E+01	3.818470E+01	5.983420E+01	4.070460E+01	**3.292440E+01**	3.865720E+01
f_{17}	Rank	2	7	3	4	8	6	**1**	5
	Median	1.261932E+02	3.562794E+02	**9.039060E+01**	1.482849E+02	1.883601E+02	1.999784E+02	3.194422E+02	3.202044E+02
	Rank	2	8	**1**	3	4	5	6	7
	(Std Dev)	(8.375947E+01)	(2.259472E+02)	(6.169525E+01)	(1.043121E+02)	(1.126427E+02)	(1.534478E+02)	(1.831913E+02)	(1.854067E+02)
	Mean	**1.688904E+02**	2.628183E+04	2.519472E+05	2.129810E+05	3.765127E+04	1.401139E+05	3.768891E+04	3.103803E+05
	Rank|WRST	**1**	2 | >	7 | >	6 | >	3 | >	5 | >	4 | >	8 | >
	Min	**5.812380E+01**	1.252548E+03	2.827476E+04	3.566825E+04	3.662660E+02	3.341797E+04	7.536985E+02	2.614612E+04
f_{18}	Rank	**1**	4	6	8	2	7	3	5
	Median	**1.555854E+02**	1.882396E+04	1.781868E+05	1.673196E+05	1.329496E+04	1.255877E+05	1.983603E+04	1.960550E+05
	Rank	**1**	3	7	6	2	5	4	8
	(Std Dev)	(8.402342E+01)	(2.384504E+04)	(2.443267E+05)	(1.459379E+05)	(4.683610E+04)	(8.164271E+04)	(8.676745E+04)	(2.801939E+05)
	Mean	**8.703206E+01**	1.310630E+04	9.398490E+03	2.773776E+03	5.963198E+03	3.900652E+03	4.033422E+03	6.721717E+03
	Rank|WRST	**1**	8 | >	7 | >	2 | >	5 | >	3 | >	4 | >	6 | >
	Min	1.687720E+01	2.691660E+01	**9.529800E+00**	3.003950E+01	1.368640E+01	3.428160E+01	6.208920E+01	1.400627E+02
f_{19}	Rank	3	4	**1**	5	2	6	7	8
	Median	**4.530610E+01**	1.040744E+04	5.729112E+03	1.730570E+03	2.763494E+03	3.402879E+03	3.330922E+03	3.070736E+03
	Rank	**1**	8	7	2	3	6	5	4
	(Std Dev)	(1.225526E+02)	(1.308854E+04)	(1.012378E+04)	(3.156671E+03)	(7.813419E+03)	(3.022542E+03)	(4.210242E+03)	(8.043860E+03)
	Mean	1.724088E+02	3.814761E+02	**1.117255E+02**	1.599922E+02	2.324817E+02	3.236438E+02	3.105722E+02	3.417379E+02
	Rank|WRST	3	8 | >	**1** | <	2 | =	4 | >	6 | >	5 | >	7 | >
	Min	3.115910E+01	4.947890E+01	**7.125600E+00**	2.846790E+01	6.901100E+01	9.288840E+01	4.085660E+01	7.899460E+01
f_{20}	Rank	3	5	**1**	2	6	8	4	7
	Median	1.772095E+02	3.588702E+02	**7.300850E+01**	1.643838E+02	2.220688E+02	2.613728E+02	2.986910E+02	3.152940E+02
	Rank	3	8	**1**	2	4	5	6	7
	(Std Dev)	(7.149171E+01)	(2.062748E+02)	(8.920745E+01)	(4.839412E+01)	(1.026383E+02)	(1.590403E+02)	(1.436799E+02)	(1.494602E+02)
	Mean	**2.130154E+02**	2.691493E+02	2.375482E+02	2.286103E+02	2.320400E+02	2.254402E+02	2.590706E+02	2.958304E+02
	Rank|WRST	**1**	7 | >	5 | >	3 | >	4 | >	2 | >	6 | >	8 | >
	Min	**1.000000E+02**	2.339698E+02	2.093167E+02	2.056614E+02	2.132796E+02	2.127057E+02	2.187653E+02	2.388027E+02
f_{21}	Rank	**1**	7	3	2	5	4	6	8
	Median	**2.165577E+02**	2.676708E+02	2.206449E+02	2.209191E+02	2.321826E+02	2.256177E+02	2.581438E+02	2.944781E+02
	Rank	**1**	7	2	3	5	4	6	8
	(Std Dev)	(2.364085E+01)	(2.042138E+01)	(3.188657E+01)	(2.233116E+01)	(9.712055E+00)	(5.547758E+00)	(1.378619E+01)	(2.800052E+01)
	Mean	1.561555E+02	2.269891E+03	4.846610E+02	**1.000000E+02**	2.249476E+02	1.003486E+02	3.340026E+02	8.944870E+02
	Rank|WRST	3	8 | >	6 | >	**1** | =	4 | =	2 | =	5 | >	7 | >
	Min	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**
f_{22}	Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Median	**1.000000E+02**	2.875676E+03	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	1.042105E+02	**1.000000E+02**
	Rank	**1**	8	**1**	**1**	**1**	**1**	7	**1**
	(Std Dev)	(4.010305E+02)	(1.640818E+03)	(1.187645E+03)	(0.000000E+00)	(6.273120E+02)	(9.969358E-01)	(9.311593E+02)	(1.474063E+03)
	Mean	**3.701966E+02**	4.116898E+02	3.926510E+02	3.895633E+02	3.866387E+02	3.903242E+02	4.405769E+02	4.985322E+02
	Rank|WRST	**1**	6 | >	5 | >	3 | >	2 | >	4 | >	7 | >	8 | >
	Min	**3.559407E+02**	3.765616E+02	3.573484E+02	3.602194E+02	3.629958E+02	3.683546E+02	3.921460E+02	4.021314E+02
f_{23}	Rank	**1**	6	2	3	4	5	7	8
	Median	**3.712761E+02**	4.134354E+02	3.768138E+02	3.814173E+02	3.864180E+02	3.876183E+02	4.304710E+02	4.856594E+02
	Rank	**1**	6	2	3	4	5	7	8
	(Std Dev)	(6.468049E+00)	(1.975260E+01)	(3.277077E+01)	(2.708143E+01)	(1.122535E+01)	(1.453301E+01)	(3.663767E+01)	(5.199236E+01)
	Mean	**4.134462E+02**	4.899419E+02	4.694986E+02	4.505862E+02	4.547978E+02	4.563117E+02	5.050632E+02	5.892920E+02
	Rank|WRST	**1**	6 | >	5 | >	2 | >	3 | >	4 | >	7 | >	8 | >
f_{24}									

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Min	**2.000000E+02**	4.450524E+02	4.291791E+02	4.257623E+02	4.347075E+02	4.343221E+02	4.456714E+02	4.789601E+02
	Rank	**1**	6	3	2	5	4	7	8
	Median	**4.354177E+02**	4.905134E+02	4.430306E+02	4.429797E+02	4.532911E+02	4.556908E+02	4.965522E+02	5.869986E+02
	Rank	**1**	6	3	2	4	5	7	8
	(Std Dev)	(7.129830E+01)	(2.208172E+01)	(3.931808E+01)	(2.365409E+01)	(1.046405E+01)	(1.176167E+01)	(4.044869E+01)	(6.983721E+01)
f_{25}	Mean	3.876079E+02	3.866369E+02	3.943891E+02	**3.790023E+02**	3.985556E+02	3.921852E+02	4.129943E+02	4.174918E+02
	Rank|WRST	3	2 | =	5 | >	**1** | <	6 | >	4 | >	7 | >	8 | >
	Min	3.834277E+02	3.834109E+02	3.887036E+02	**3.782936E+02**	3.844508E+02	3.836995E+02	3.839586E+02	3.868155E+02
	Rank	3	2	8	**1**	6	4	5	7
	Median	3.870054E+02	3.870003E+02	3.918692E+02	**3.786184E+02**	3.929227E+02	3.887367E+02	4.028746E+02	4.181733E+02
	Rank	3	2	5	**1**	6	4	7	8
	(Std Dev)	(6.229457E+00)	(1.189773E+00)	(6.363114E+00)	(9.897323E-01)	(1.232610E+01)	(1.008312E+01)	(2.717043E+01)	(1.813891E+01)
f_{26}	Mean	**3.495867E+02**	1.604770E+03	1.272836E+03	1.184079E+03	1.322010E+03	1.461181E+03	1.643466E+03	2.025905E+03
	Rank|WRST	**1**	6 | >	3 | >	2 | >	4 | >	5 | >	7 | >	8 | >
	Min	**2.000000E+02**	**2.000000E+02**	9.593049E+02	**2.000000E+02**	**2.000000E+02**	**2.000000E+02**	**2.000000E+02**	**2.000000E+02**
	Rank	**1**	**1**	8	**1**	**1**	**1**	**1**	**1**
	Median	**3.000000E+02**	1.663680E+03	1.173292E+03	1.178135E+03	1.346035E+03	1.517643E+03	1.718706E+03	2.424315E+03
	Rank	**1**	6	2	3	4	5	7	8
	(Std Dev)	(2.884979E+02)	(4.396993E+02)	(3.323863E+02)	(2.352306E+02)	(3.094289E+02)	(3.821245E+02)	(1.291945E+03)	(1.094023E+03)
f_{27}	Mean	5.099578E+02	5.169569E+02	5.220893E+02	**4.853770E+02**	5.349945E+02	5.335280E+02	5.392567E+02	6.458385E+02
	Rank|WRST	2	3 | >	4 | >	**1** | <	6 | >	5 | >	7 | >	8 | >
	Min	4.906652E+02	4.956169E+02	5.047742E+02	**4.491095E+02**	5.000071E+02	4.992067E+02	5.100308E+02	5.659961E+02
	Rank	2	3	6	**1**	5	4	7	8
	Median	5.102593E+02	5.142328E+02	5.219866E+02	**5.000064E+02**	5.303570E+02	5.326764E+02	5.382906E+02	6.353997E+02
	Rank	2	3	4	**1**	5	6	7	8
	(Std Dev)	(1.055004E+01)	(1.377164E+01)	(8.020770E+00)	(2.054829E+01)	(2.996176E+01)	(1.681703E+01)	(1.439111E+01)	(4.670726E+01)
f_{28}	Mean	**3.060756E+02**	3.392997E+02	4.392948E+02	4.053297E+02	4.058634E+02	3.966689E+02	4.950095E+02	4.461367E+02
	Rank|WRST	**1**	2 | >	6 | >	4 | >	5 | >	3 | >	8 | >	7 | >
	Min	**3.000000E+02**	**3.000000E+02**	**3.000000E+02**	3.140285E+02	**3.000000E+02**	**3.000000E+02**	4.177236E+02	3.011479E+02
	Rank	**1**	**1**	**1**	7	**1**	**1**	8	6
	Median	**3.000000E+02**	**3.000000E+02**	4.466757E+02	4.041709E+02	4.071991E+02	4.047307E+02	4.937322E+02	4.402401E+02
	Rank	**1**	**1**	7	3	5	4	8	6
	(Std Dev)	(2.454434E+01)	(5.708749E+01)	(3.951984E+01)	(2.210217E+01)	(5.726847E+01)	(3.417800E+01)	(3.449953E+01)	(5.470662E+01)
f_{29}	Mean	5.762106E+02	8.189940E+02	5.230041E+02	**5.177576E+02**	7.579059E+02	7.250974E+02	7.028659E+02	8.948919E+02
	Rank|WRST	3	7 | >	2 | <	**1** | <	6 | >	5 | >	4 | >	8 | >
	Min	4.112461E+02	5.174793E+02	**3.752203E+02**	3.767935E+02	5.343136E+02	4.775942E+02	4.492997E+02	5.608371E+02
	Rank	3	6	**1**	2	7	5	4	8
	Median	5.406649E+02	8.278428E+02	**5.034346E+02**	5.061645E+02	7.269571E+02	6.845853E+02	6.894951E+02	8.605693E+02
	Rank	3	7	**1**	2	6	4	5	8
	(Std Dev)	(1.015197E+02)	(1.930054E+02)	(8.075580E+01)	(1.107433E+02)	(1.685417E+02)	(1.714031E+02)	(1.808463E+02)	(2.114720E+02)
f_{30}	Mean	3.744833E+03	5.819768E+03	4.497355E+03	**1.991400E+03**	1.642278E+04	4.864679E+03	3.707635E+03	2.719504E+04
	Rank|WRST	3	6 | >	4 | >	**1** | <	7 | >	5 | >	2 | =	8 | >
	Min	2.592887E+03	2.063031E+03	2.851280E+03	**2.281946E+02**	1.552110E+03	2.546369E+03	2.103177E+03	4.734106E+03
	Rank	6	3	7	**1**	2	5	4	8
	Median	3.709775E+03	4.383129E+03	4.280878E+03	**1.284874E+03**	1.209858E+04	4.280535E+03	3.433418E+03	1.836285E+04
	Rank	3	6	5	**1**	7	4	2	8
	(Std Dev)	(6.335384E+02)	(3.335561E+03)	(1.388514E+03)	(2.052406E+03)	(1.248893E+04)	(1.946332E+03)	(1.103597E+03)	(2.824835E+04)

Table 10: Experimental results for PSO-based variants on 50-dimensional CEC2017 benchmark functions based on 51 runs.

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
f_1	Mean	**0.000000E+00**	7.076403E+03	5.141341E+03	2.085471E+03	3.266516E+03	3.223689E+03	3.034080E+08	3.860011E+03
	Rank|WRST	**1**	7 | >	6 | >	2 | >	4 | >	3 | >	8 | >	5 | >
	Min	**0.000000E+00**	2.812960E+00	3.290800E-01	1.267790E+00	1.215733E+01	1.087970E+00	5.262206E+03	4.157500E-01
	Rank	**1**	6	2	5	7	4	8	3
	Median	**0.000000E+00**	4.050789E+03	2.965203E+03	1.086200E+03	1.874394E+03	1.614313E+03	6.423342E+07	3.193664E+03
	Rank	**1**	7	5	2	4	3	8	6
	(Std Dev)	(0.000000E+00)	(8.009123E+03)	(5.771259E+03)	(3.069071E+03)	(4.522605E+03)	(4.419410E+03)	(5.286449E+08)	(4.046057E+03)
f_2	Mean	5.074570E+20	**3.976008E+05**	9.346504E+40	7.055374E+37	1.114766E+41	4.852097E+37	6.061847E+17	3.246290E+43
	Rank|WRST	3	**1** | <	6 | >	5 | <	7 | >	4 | >	2 | <	8 | >
	Min	4.397028E+18	5.000000E+00	3.337230E+03	**3.090000E-03**	1.796790E+24	8.077790E+21	7.511390E+05	2.213524E+19
	Rank	5	2	3	**1**	8	7	4	6
	Median	1.642765E+20	**2.000000E+02**	9.561918E+28	3.666035E+09	1.761801E+32	5.687417E+30	2.975808E+10	1.675148E+30
	Rank	4	**1**	5	2	8	7	3	6
	(Std Dev)	(1.285972E+21)	(2.796587E+06)	(6.645597E+41)	(4.532474E+38)	(6.982359E+41)	(3.103194E+38)	(2.620470E+18)	(2.317551E+44)
f_3	Mean	**0.000000E+00**	2.801759E-01	9.957364E+03	1.850763E+04	1.906886E+02	7.233679E+04	1.325243E+01	3.625309E+04
	Rank|WRST	**1**	2 | >	5 | >	6 | >	4 | >	8 | >	3 | >	7 | >
	Min	**0.000000E+00**	**0.000000E+00**	1.276230E+03	8.714081E+03	5.171940E+00	4.152285E+04	1.631100E-01	1.873920E+04
	Rank	**1**	**1**	5	6	4	8	3	7
	Median	**0.000000E+00**	**0.000000E+00**	9.274396E+03	1.722861E+04	6.933962E+01	6.873782E+04	4.169930E+00	3.674030E+04
	Rank	**1**	**1**	5	6	4	8	3	7
	(Std Dev)	(0.000000E+00)	(1.187668E+00)	(5.045154E+03)	(5.208687E+03)	(3.506950E+02)	(2.105171E+04)	(1.941874E+01)	(7.982792E+03)
f_4	Mean	**1.652989E+01**	3.263276E+01	2.480251E+02	8.592004E+01	2.366488E+02	1.151387E+02	2.566787E+02	2.820861E+02
	Rank|WRST	**1**	2 | >	6 | >	3 | >	5 | >	4 | >	7 | >	8 | >

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Min	**0.000000E+00**	1.200000E-04	1.417731E+02	2.307305E+01	8.910110E+01	2.923024E+01	6.525000E-02	1.346084E+02
	Rank	**1**	2	8	4	6	5	3	7
	Median	**3.986620E+00**	8.003760E+00	2.499894E+02	9.676852E+01	2.432806E+02	1.015505E+02	1.471587E+02	2.515045E+02
	Rank	**1**	2	7	3	6	4	5	8
	(Std Dev)	(2.200168E+01)	(4.593453E+01)	(2.415775E+01)	(3.714774E+01)	(5.885502E+01)	(5.013476E+01)	(1.957288E+02)	(1.429603E+02)
f_5	Mean	**3.909602E+01**	1.404666E+02	4.164294E+01	4.057711E+01	6.876914E+01	6.032176E+01	1.529410E+02	2.397838E+02
	Rank|WRST	**1**	6 | >	3 | <	2 | <	5 | >	4 | >	7 | >	8 | >
	Min	2.686388E+01	7.860168E+01	**1.790926E+01**	2.188912E+01	3.780843E+01	4.178826E+01	9.651090E+01	1.671527E+02
	Rank	3	6	**1**	2	4	5	7	8
	Median	3.880338E+01	1.382990E+02	3.183868E+01	**2.985017E+01**	7.064204E+01	6.069244E+01	1.502385E+02	2.318243E+02
	Rank	3	6	2	**1**	5	4	7	8
	(Std Dev)	(6.384170E+00)	(3.796547E+01)	(4.113957E+01)	(3.736862E+01)	(1.655058E+01)	(1.134592E+01)	(3.049493E+01)	(4.339114E+01)
f_6	Mean	3.090204E+00	1.010004E-01	3.300471E-02	**7.843137E-07**	4.850953E-01	7.577559E-01	5.517616E-01	1.290892E+01
	Rank|WRST	7	3 | <	2 | <	**1** | <	4 | <	6 | <	5 | <	8 | >
	Min	1.299920E+00	**0.000000E+00**	6.900000E-04	**0.000000E+00**	1.100000E-04	2.112000E-02	4.200000E-04	3.499680E+00
	Rank	7	**1**	5	**1**	3	6	4	8
	Median	2.925240E+00	**0.000000E+00**	2.260000E-02	**0.000000E+00**	6.962000E-02	5.845900E-01	3.455200E-01	1.188200E+01
	Rank	7	**1**	3	**1**	4	6	5	8
	(Std Dev)	(8.827571E-01)	(3.204524E-01)	(3.725366E-02)	(2.715244E-06)	(9.628648E-01)	(6.160529E-01)	(7.109815E-01)	(5.984485E+00)
f_7	Mean	1.205140E+02	1.785451E+02	1.786987E+02	1.675127E+02	1.234101E+02	**1.088908E+02**	2.376475E+02	3.492500E+02
	Rank|WRST	2	5 | >	6 | =	4 | =	3 | =	**1** | <	7 | >	8 | >
	Min	9.341788E+01	1.208625E+02	7.626856E+01	**7.386984E+01**	9.138318E+01	8.180412E+01	1.566245E+02	2.302301E+02
	Rank	5	6	2	**1**	4	3	7	8
	Median	1.200028E+02	1.798989E+02	1.116175E+02	1.104105E+02	1.170284E+02	**1.092685E+02**	2.398265E+02	3.393787E+02
	Rank	5	6	3	2	4	**1**	7	8
	(Std Dev)	(1.605322E+01)	(2.488642E+01)	(9.952238E+01)	(9.820470E+01)	(3.036435E+01)	(1.451257E+01)	(4.634474E+01)	(6.966800E+01)
f_8	Mean	**3.909601E+01**	1.403883E+02	4.935429E+01	5.731136E+01	6.535528E+01	6.213609E+01	1.379166E+02	2.479776E+02
	Rank|WRST	**1**	7 | >	2 | <	3 | =	5 | >	4 | >	6 | >	8 | >
	Min	1.890422E+01	8.059151E+01	**1.492439E+01**	2.259800E+01	3.581851E+01	4.178826E+01	6.399946E+01	1.452634E+02
	Rank	2	7	**1**	3	4	5	6	8
	Median	3.780842E+01	1.392939E+02	**3.382859E+01**	3.383187E+01	6.467225E+01	6.367731E+01	1.392941E+02	2.557030E+02
	Rank	3	6	**1**	2	5	4	7	8
	(Std Dev)	(8.977350E+00)	(3.087290E+01)	(5.015543E+01)	(6.239860E+01)	(1.880942E+01)	(1.011204E+01)	(3.317264E+01)	(5.354995E+01)
f_9	Mean	3.637730E+01	1.197030E+03	1.164394E+01	**7.269829E-01**	1.254758E+01	2.750835E+02	9.503619E+02	3.942157E+03
	Rank|WRST	4	7 | >	2 | <	**1** | <	3 | <	5 | >	6 | >	8 | >
	Min	2.061420E+00	4.993320E+00	6.334500E-01	**0.000000E+00**	5.438500E-01	6.521226E+01	2.562580E+02	1.173989E+03
	Rank	4	5	3	**1**	2	6	7	8
	Median	2.091665E+01	4.339420E+02	8.059120E+00	**5.438500E-01**	8.394820E+00	2.186336E+02	8.631040E+02	3.630809E+03
	Rank	4	6	2	**1**	3	5	7	8
	(Std Dev)	(4.014715E+01)	(2.055486E+03)	(1.293148E+01)	(7.433190E-01)	(1.298767E+01)	(2.013943E+02)	(6.401494E+02)	(2.163078E+03)
f_{10}	Mean	4.786255E+03	5.496636E+03	5.955452E+03	5.450805E+03	5.678718E+03	**4.253842E+03**	5.886075E+03	6.051639E+03
	Rank|WRST	2	4 | >	7 | >	3 | =	5 | >	**1** | <	6 | >	8 | >
	Min	3.617992E+03	3.536680E+03	2.978732E+03	3.120530E+03	4.023845E+03	**2.525854E+03**	4.554162E+03	4.503133E+03
	Rank	5	4	2	3	6	**1**	8	7
	Median	4.872800E+03	5.472142E+03	5.992501E+03	4.733896E+03	5.615625E+03	**4.257793E+03**	5.771557E+03	6.163760E+03
	Rank	3	4	7	2	5	**1**	6	8
	(Std Dev)	(5.773028E+02)	(8.805082E+02)	(9.200883E+02)	(2.166202E+03)	(8.631444E+02)	(6.386260E+02)	(7.205047E+02)	(7.704115E+02)
f_{11}	Mean	8.518573E+01	**8.429387E+01**	1.413356E+02	1.022649E+02	2.471378E+02	1.747839E+02	2.130152E+02	2.270259E+02
	Rank|WRST	2	**1** | =	4 | >	3 | =	8 | >	5 | >	6 | >	7 | >
	Min	**2.785880E+01**	4.407680E+01	6.908670E+01	3.626480E+01	1.145677E+02	9.885730E+01	1.085045E+02	9.008780E+01
	Rank	**1**	3	4	2	8	6	7	5
	Median	**7.796220E+01**	7.895710E+01	1.368561E+02	9.341980E+01	2.421353E+02	1.645406E+02	1.947205E+02	2.016208E+02
	Rank	**1**	2	4	3	8	5	6	7
	(Std Dev)	(3.095221E+01)	(2.949733E+01)	(4.102431E+01)	(4.468418E+01)	(7.396243E+01)	(4.660492E+01)	(6.529057E+01)	(9.673336E+01)
f_{12}	Mean	**3.996475E+03**	5.248130E+04	2.132520E+06	2.293362E+06	9.439333E+05	9.849751E+05	1.320330E+07	6.394430E+06
	Rank|WRST	**1**	2 | >	5 | >	6 | >	3 | >	4 | >	8 | >	7 | >
	Min	**1.135194E+03**	1.140822E+04	2.487645E+05	1.856253E+05	4.417761E+04	4.359486E+04	2.321810E+05	1.324783E+05
	Rank	**1**	2	8	6	4	3	7	5
	Median	**2.300877E+03**	4.744157E+04	1.598878E+06	2.100076E+06	4.777731E+05	7.912633E+05	5.387594E+06	1.897197E+06
	Rank	**1**	2	5	7	3	4	8	6
	(Std Dev)	(5.312949E+03)	(2.693536E+04)	(1.730263E+06)	(1.247141E+06)	(1.367576E+06)	(7.400327E+05)	(2.585519E+07)	(1.674265E+07)
f_{13}	Mean	**1.819712E+02**	6.121069E+03	2.808415E+03	5.164419E+03	3.888529E+03	4.654024E+03	2.278112E+04	5.138762E+03
	Rank|WRST	**1**	7 | >	2 | >	6 | >	3 | >	4 | >	8 | >	5 | >
	Min	**3.883760E+01**	4.965020E+01	1.342172E+02	1.842724E+02	1.847940E+02	3.011768E+02	1.916357E+02	2.532334E+02
	Rank	**1**	2	3	4	5	8	6	7
	Median	**1.562411E+02**	3.528623E+03	1.240431E+03	2.507067E+03	2.837363E+03	2.856693E+03	4.632550E+03	2.152311E+03
	Rank	**1**	7	2	4	5	6	8	3
	(Std Dev)	(1.153694E+02)	(7.193056E+03)	(3.317002E+03)	(6.302865E+03)	(4.193201E+03)	(4.362386E+03)	(1.132147E+05)	(6.074107E+03)
f_{14}	Mean	**2.320420E+02**	4.448106E+03	8.179160E+04	6.603217E+04	1.553035E+04	5.475175E+04	3.973819E+03	1.387562E+05
	Rank|WRST	**1**	3 | >	7 | >	6 | >	4 | >	5 | >	2 | >	8 | >
	Min	**1.473559E+02**	3.245102E+02	2.517942E+03	3.066296E+03	1.986999E+02	3.979470E+03	6.093767E+02	4.667031E+03
	Rank	**1**	3	5	6	2	7	4	8
	Median	**2.179974E+02**	3.775224E+03	8.525985E+04	5.512240E+04	2.308592E+03	4.780150E+04	3.079231E+03	4.211052E+04
	Rank	**1**	4	8	7	2	6	3	5
	(Std Dev)	(5.384711E+01)	(2.799529E+03)	(5.763749E+04)	(6.762553E+04)	(3.659831E+04)	(4.013575E+04)	(3.274996E+03)	(1.871911E+05)
f_{15}	Mean	**6.508189E+01**	9.388169E+03	3.983471E+03	4.421270E+03	3.731288E+03	5.215785E+03	9.531062E+03	7.877467E+03

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Rank|WRST	**1**	7 | >	3 | >	4 | >	2 | >	5 | >	8 | >	6 | >
	Min	**2.583230E+01**	3.863280E+01	6.536950E+01	1.209738E+02	3.582240E+01	2.480725E+02	3.452024E+02	1.648494E+02
	Rank	**1**	3	4	5	2	7	8	6
	Median	**4.835680E+01**	7.155327E+03	2.896738E+03	2.460278E+03	2.421932E+03	4.339759E+03	9.114026E+03	3.019479E+03
	Rank	**1**	7	4	3	2	6	8	5
	(Std Dev)	(9.129032E+01)	(8.734007E+03)	(3.516612E+03)	(4.751168E+03)	(3.678175E+03)	(4.075513E+03)	(5.951156E+03)	(1.128688E+04)
	Mean	8.844866E+02	1.534087E+03	1.049902E+03	**6.177373E+02**	1.372087E+03	9.439766E+02	1.465828E+03	1.719244E+03
	Rank|WRST	2	7 | >	4 | =	**1** | <	5 | >	3 | =	6 | >	8 | >
	Min	**1.354178E+02**	9.288969E+02	2.278637E+02	1.390888E+02	5.208664E+02	3.712918E+02	5.883831E+02	7.071826E+02
f_{16}	Rank	**1**	8	3	2	5	4	6	7
	Median	8.189479E+02	1.546208E+03	1.057317E+03	**5.933513E+02**	1.416011E+03	8.539921E+02	1.488323E+03	1.728474E+03
	Rank	2	7	4	**1**	5	3	6	8
	(Std Dev)	(3.402577E+02)	(3.574978E+02)	(5.318279E+02)	(3.016599E+02)	(3.740255E+02)	(3.523113E+02)	(3.286328E+02)	(3.739991E+02)
	Mean	8.674605E+02	1.043841E+03	9.370026E+02	**6.429132E+02**	1.107029E+03	1.001325E+03	1.001940E+03	1.188070E+03
	Rank|WRST	2	6 | >	3 | =	**1** | <	7 | >	4 | >	5 | >	8 | >
	Min	3.482257E+02	4.490823E+02	**1.945665E+02**	2.716961E+02	4.976170E+02	4.143295E+02	2.431622E+02	5.568219E+02
f_{17}	Rank	4	6	**1**	3	7	5	2	8
	Median	8.437954E+02	1.045632E+03	1.006257E+03	**6.621637E+02**	1.091539E+03	9.671770E+02	1.020295E+03	1.180390E+03
	Rank	2	6	4	**1**	7	3	5	8
	(Std Dev)	(2.487628E+02)	(2.867384E+02)	(3.640004E+02)	(2.037159E+02)	(2.498705E+02)	(3.014869E+02)	(3.073889E+02)	(2.764881E+02)
	Mean	**2.020411E+02**	2.641331E+04	7.548194E+05	9.115363E+05	5.008608E+05	2.777679E+05	6.949851E+04	2.203766E+06
	Rank|WRST	**1**	2 | >	6 | >	7 | >	5 | >	4 | >	3 | >	8 | >
	Min	**7.515190E+01**	7.917570E+03	7.336196E+04	1.215538E+05	6.749911E+03	7.527775E+04	8.564579E+03	2.834907E+05
f_{18}	Rank	**1**	3	5	7	2	6	4	8
	Median	**1.786919E+02**	2.302647E+04	4.910116E+05	7.861303E+05	1.661425E+05	2.446955E+05	6.016825E+04	1.573534E+06
	Rank	**1**	2	6	7	4	5	3	8
	(Std Dev)	(7.859069E+01)	(1.392129E+04)	(5.818310E+05)	(4.972166E+05)	(7.109434E+05)	(1.242456E+05)	(4.423948E+04)	(1.970376E+06)
	Mean	**2.764872E+02**	1.667224E+04	7.714599E+03	1.528804E+04	1.361729E+04	1.395210E+04	1.569431E+04	1.475342E+04
	Rank|WRST	**1**	8 | >	2 | >	6 | >	3 | >	4 | >	7 | >	5 | >
	Min	3.647960E+01	4.323650E+01	**2.086090E+01**	1.412908E+03	7.842350E+01	8.822039E+02	5.606076E+02	1.613793E+02
f_{19}	Rank	2	3	**1**	8	4	7	6	5
	Median	**2.133755E+02**	1.419365E+04	7.445941E+03	1.331668E+04	1.127366E+04	1.248573E+04	1.512866E+04	1.012351E+04
	Rank	**1**	7	2	6	4	5	8	3
	(Std Dev)	(2.449418E+02)	(1.390876E+04)	(6.382980E+03)	(8.685536E+03)	(1.084402E+04)	(7.955016E+03)	(7.059313E+03)	(1.500541E+04)
	Mean	6.208777E+02	1.013627E+03	6.851220E+02	**2.571127E+02**	6.940470E+02	5.885104E+02	7.293609E+02	8.263827E+02
	Rank|WRST	3	8 | >	4 | =	**1** | <	5 | =	2 | =	6 | >	7 | >
	Min	1.869110E+02	1.655446E+02	8.536400E+01	**4.794150E+01**	1.520854E+02	1.166933E+02	6.014210E+01	2.236764E+02
f_{20}	Rank	7	6	3	**1**	5	4	2	8
	Median	6.040714E+02	1.052236E+03	7.070911E+02	**2.159428E+02**	6.936383E+02	6.183819E+02	7.307466E+02	8.325884E+02
	Rank	2	8	5	**1**	4	3	6	7
	(Std Dev)	(2.111587E+02)	(3.436591E+02)	(2.409182E+02)	(1.686284E+02)	(2.412459E+02)	(2.621278E+02)	(3.214105E+02)	(2.996793E+02)
	Mean	**2.425725E+02**	3.343113E+02	2.533120E+02	2.498565E+02	2.701410E+02	2.607043E+02	3.381668E+02	4.008768E+02
	Rank|WRST	**1**	6 | >	3 | <	2 | <	5 | >	4 | >	7 | >	8 | >
	Min	2.288101E+02	2.810065E+02	2.208927E+02	**2.167389E+02**	2.382234E+02	2.362749E+02	2.874733E+02	3.198239E+02
f_{21}	Rank	3	6	2	**1**	5	4	7	8
	Median	2.403144E+02	3.255349E+02	2.363531E+02	**2.306817E+02**	2.703512E+02	2.610743E+02	3.367986E+02	3.911892E+02
	Rank	3	6	2	**1**	5	4	7	8
	(Std Dev)	(9.485986E+00)	(3.491248E+01)	(4.876756E+01)	(5.310598E+01)	(1.555821E+01)	(1.364125E+01)	(2.886686E+01)	(4.666716E+01)
	Mean	3.200991E+03	6.356237E+03	6.300026E+03	2.669663E+03	5.380035E+03	**1.366630E+03**	4.771007E+03	6.469614E+03
	Rank|WRST	3	7 | >	6 | >	2 | =	5 | >	**1** | <	4 | >	8 | >
	Min	**1.000000E+02**	3.528669E+03	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	**1.000000E+02**	1.000596E+02	**1.000000E+02**
f_{22}	Rank	**1**	8	**1**	**1**	**1**	**1**	7	**1**
	Median	4.676148E+03	6.461185E+03	6.580293E+03	3.229430E+03	5.868608E+03	**1.000000E+02**	6.189104E+03	7.071847E+03
	Rank	3	6	7	2	4	**1**	5	8
	(Std Dev)	(2.467572E+03)	(9.507339E+02)	(2.016309E+03)	(2.614575E+03)	(2.169999E+03)	(2.177651E+03)	(2.986358E+03)	(2.049114E+03)
	Mean	**4.733980E+02**	5.612327E+02	4.931848E+02	5.103823E+02	5.180367E+02	5.339020E+02	6.279327E+02	7.394313E+02
	Rank|WRST	**1**	6 | >	2 | =	3 | >	4 | >	5 | >	7 | >	8 | >
	Min	4.521144E+02	4.780991E+02	**4.423187E+02**	4.437270E+02	4.544600E+02	4.813426E+02	5.211086E+02	5.763074E+02
f_{23}	Rank	3	5	**1**	2	4	6	7	8
	Median	**4.713340E+02**	5.622488E+02	4.800744E+02	4.879858E+02	5.137586E+02	5.294853E+02	6.207313E+02	7.402871E+02
	Rank	**1**	6	2	3	4	5	7	8
	(Std Dev)	(1.268818E+01)	(3.439806E+01)	(5.461954E+01)	(6.942848E+01)	(3.102846E+01)	(3.594463E+01)	(7.298629E+01)	(6.379280E+01)
	Mean	**5.340279E+02**	6.303252E+02	5.972233E+02	5.644362E+02	5.704692E+02	6.015499E+02	6.649288E+02	8.771360E+02
	Rank|WRST	**1**	6 | >	4 | >	2 | >	3 | >	5 | >	7 | >	8 | >
	Min	**5.173473E+02**	5.748158E+02	5.192175E+02	5.242209E+02	5.323998E+02	5.515132E+02	5.552061E+02	6.362588E+02
f_{24}	Rank	**1**	7	2	3	4	5	6	8
	Median	**5.321350E+02**	6.278887E+02	5.531050E+02	5.479168E+02	5.684434E+02	5.968161E+02	6.484992E+02	8.589091E+02
	Rank	**1**	6	3	2	4	5	7	8
	(Std Dev)	(9.323682E+00)	(3.045947E+01)	(8.080759E+01)	(4.451768E+01)	(1.968372E+01)	(3.032183E+01)	(7.140282E+01)	(1.161773E+02)
	Mean	4.921153E+02	5.231984E+02	5.912448E+02	**4.389681E+02**	6.469177E+02	5.602258E+02	7.211006E+02	7.239120E+02
	Rank|WRST	2	3 | >	5 | >	**1** | <	6 | >	4 | >	7 | >	8 | >
	Min	**4.283679E+02**	4.605128E+02	5.583471E+02	4.311367E+02	5.628041E+02	4.618454E+02	5.889962E+02	5.485181E+02
f_{25}	Rank	**1**	3	6	2	7	4	8	5
	Median	4.803157E+02	5.272063E+02	5.947913E+02	**4.312053E+02**	6.442096E+02	5.642895E+02	7.060436E+02	7.129755E+02
	Rank	2	3	5	**1**	6	4	7	8
	(Std Dev)	(2.995282E+01)	(3.960667E+01)	(1.489181E+01)	(2.026065E+01)	(3.803897E+01)	(3.032802E+01)	(7.962217E+01)	(8.348749E+01)

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
f_{26}	Mean	**6.818628E+02**	2.552703E+03	1.616989E+03	1.461736E+03	1.858336E+03	2.232882E+03	3.064783E+03	3.093891E+03
	Rank|WRST	**1**	6 | >	3 | >	2 | >	4 | >	5 | >	7 | >	8 | >
	Min	**3.000000E+02**	1.597776E+03	1.254898E+03	**3.000000E+02**	1.422043E+03	1.594977E+03	3.286493E+02	**3.000000E+02**
	Rank	**1**	8	5	**1**	6	7	4	**1**
	Median	**3.000001E+02**	2.505684E+03	1.576784E+03	1.361431E+03	1.814497E+03	2.231505E+03	2.603185E+03	3.879908E+03
	Rank	**1**	6	3	2	4	5	7	8
	(Std Dev)	(5.741020E+02)	(3.930881E+02)	(2.770009E+02)	(4.272582E+02)	(2.445500E+02)	(2.997245E+02)	(2.034778E+03)	(2.394975E+03)
f_{27}	Mean	5.689737E+02	6.408263E+02	6.304876E+02	**5.009275E+02**	8.692791E+02	7.812562E+02	7.694693E+02	1.190919E+03
	Rank|WRST	2	4 | >	3 | >	**1** | <	7 | >	6 | >	5 | >	8 | >
	Min	5.241784E+02	5.331133E+02	5.519316E+02	**4.533485E+02**	6.546704E+02	6.676490E+02	5.970506E+02	8.053267E+02
	Rank	2	3	4	**1**	6	7	5	8
	Median	5.573874E+02	6.371716E+02	6.238902E+02	**5.000114E+02**	8.685080E+02	7.769783E+02	7.606774E+02	1.215968E+03
	Rank	2	4	3	**1**	7	6	5	8
	(Std Dev)	(3.936830E+01)	(7.272299E+01)	(4.644842E+01)	(1.995196E+01)	(1.050244E+02)	(7.303044E+01)	(1.102237E+02)	(1.582079E+02)
f_{28}	Mean	**4.660318E+02**	4.829411E+02	5.344923E+02	4.867351E+02	6.307295E+02	5.073763E+02	8.604620E+02	7.320621E+02
	Rank|WRST	**1**	2 | =	5 | >	3 | >	6 | >	4 | >	8 | >	7 | >
	Min	4.588487E+02	4.533516E+02	4.942329E+02	**4.436418E+02**	5.000121E+02	4.588508E+02	5.905139E+02	5.322729E+02
	Rank	3	2	5	**1**	6	4	8	7
	Median	**4.588487E+02**	**4.588487E+02**	5.311028E+02	5.000111E+02	6.330313E+02	5.089968E+02	8.207910E+02	7.019361E+02
	Rank	**1**	**1**	5	3	6	4	8	7
	(Std Dev)	(1.536594E+01)	(3.310178E+01)	(2.637999E+01)	(1.968681E+01)	(5.378136E+01)	(1.419027E+01)	(1.960606E+02)	(1.400389E+02)
f_{29}	Mean	8.871036E+02	1.020032E+03	6.631185E+02	**6.130687E+02**	9.206765E+02	1.145710E+03	1.125348E+03	1.497733E+03
	Rank|WRST	3	5 | >	2 | <	**1** | <	4 | =	7 | >	6 | >	8 | >
	Min	5.436342E+02	4.863711E+02	3.702028E+02	**3.129160E+02**	4.039823E+02	5.347353E+02	6.224298E+02	7.203959E+02
	Rank	6	4	2	**1**	3	5	7	8
	Median	8.906177E+02	9.869567E+02	**5.682069E+02**	5.835572E+02	9.650047E+02	1.133333E+03	1.116118E+03	1.515937E+03
	Rank	3	5	**1**	2	4	7	6	8
	(Std Dev)	(1.880332E+02)	(2.911261E+02)	(2.880957E+02)	(2.420518E+02)	(2.944212E+02)	(2.833575E+02)	(2.846112E+02)	(3.796990E+02)
f_{30}	Mean	5.994307E+05	8.180780E+05	1.254846E+06	**1.671496E+03**	4.128081E+06	9.822421E+05	9.017356E+05	1.101697E+07
	Rank|WRST	2	3 | >	6 | >	**1** | <	7 | >	5 | >	4 | >	8 | >
	Min	5.795398E+05	6.301698E+05	8.091836E+05	**3.089434E+02**	5.146396E+05	7.263161E+05	7.007398E+05	4.977497E+06
	Rank	3	4	7	**1**	2	6	5	8
	Median	5.801799E+05	8.130495E+05	1.174104E+06	**1.139253E+03**	3.058520E+06	9.324584E+05	8.838193E+05	9.565390E+06
	Rank	2	3	6	**1**	7	5	4	8
	(Std Dev)	(2.771337E+04)	(1.143923E+05)	(3.447991E+05)	(1.266849E+03)	(3.356612E+06)	(1.994163E+05)	(1.435437E+05)	(5.241668E+06)

Table 11: Experimental results for PSO-based variants on 100-dimensional CEC2017 benchmark functions based on 51 runs.

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
f_1	Mean	**0.000000E+00**	7.597632E+03	8.368468E+04	6.862729E+03	5.328571E+03	4.415628E+03	5.193177E+09	7.543372E+03
	Rank|WRST	**1**	6 | >	7 | >	4 | >	3 | >	2 | >	8 | >	5 | >
	Min	**0.000000E+00**	2.818970E+00	1.001370E+00	4.963500E-01	2.103961E+01	9.335130E+00	2.713204E+08	1.312700E+00
	Rank	**1**	5	3	2	7	6	8	4
	Median	**0.000000E+00**	5.151243E+03	5.311043E+03	5.620294E+03	2.392388E+03	2.343140E+03	4.238793E+09	3.759878E+03
	Rank	**1**	5	6	7	3	2	8	4
	(Std Dev)	(0.000000E+00)	(8.634905E+03)	(3.153003E+05)	(5.719378E+03)	(6.313674E+03)	(5.492657E+03)	(4.756088E+09)	(8.948693E+03)
f_2	Mean	1.714769E+104	8.664596E+58	1.064327E+92	6.747423E+101	1.613615E+106	1.084453E+98	**1.089319E+56**	1.502748E+113
	Rank|WRST	6	2 | =	3 | >	5 | >	7 | >	4 | >	**1** | >	8 | >
	Min	1.233019E+20	**4.090537E+06**	3.779078E+37	9.943197E+06	2.283315E+76	8.943884E+55	1.369377E+37	1.440140E+67
	Rank	3	**1**	5	2	8	6	4	7
	Median	1.747607E+22	**1.227015E+21**	6.354214E+78	1.413587E+29	6.013567E+94	3.055061E+70	1.945670E+44	1.023037E+89
	Rank	2	**1**	6	3	8	5	4	7
	(Std Dev)	(1.224590E+105)	(6.166559E+59)	(7.407448E+92)	(4.818624E+102)	(1.109160E+107)	(7.744543E+98)	(7.773780E+56)	(1.073176E+114)
f_3	Mean	**6.784314E-05**	7.724834E+04	1.034846E+05	1.552593E+05	9.677094E+03	2.949663E+05	1.034789E+04	1.729600E+05
	Rank|WRST	**1**	4 | >	5 | >	6 | >	2 | >	8 | >	3 | >	7 | >
	Min	**6.000000E-05**	2.444833E+04	6.464655E+04	1.021704E+05	3.242852E+03	2.204341E+05	1.917933E+03	1.314469E+05
	Rank	**1**	4	5	6	3	8	2	7
	Median	**7.000000E-05**	6.681792E+04	1.036540E+05	1.509015E+05	8.529070E+03	2.932169E+05	8.996225E+03	1.730882E+05
	Rank	**1**	4	5	6	2	8	3	7
	(Std Dev)	(4.153902E-06)	(4.173550E+04)	(2.065538E+04)	(2.381232E+04)	(4.886081E+03)	(4.185235E+04)	(7.146072E+03)	(1.703231E+04)
f_4	Mean	**6.564800E+01**	1.219665E+02	4.236492E+02	2.193971E+02	4.774856E+02	2.329261E+02	1.731266E+03	9.405723E+02
	Rank|WRST	**1**	2 | >	5 | >	3 | >	6 | >	4 | >	8 | >	7 | >
	Min	**0.000000E+00**	1.366100E-01	3.387727E+02	9.651967E+01	2.173780E+02	1.554416E+02	4.669812E+02	4.017101E+02
	Rank	**1**	2	6	3	5	4	8	7
	Median	**3.986620E+00**	1.259626E+02	4.202445E+02	2.168668E+02	4.913889E+02	2.314833E+02	1.530047E+03	8.404016E+02
	Rank	**1**	2	5	3	6	4	8	7
	(Std Dev)	(9.288827E+01)	(6.298308E+01)	(4.331713E+01)	(4.198751E+01)	(1.156799E+02)	(3.818985E+01)	(1.112518E+03)	(4.302328E+02)
f_5	Mean	1.324659E+02	3.558912E+02	1.151073E+02	**8.831189E+01**	1.993671E+02	2.014105E+02	4.476244E+02	6.940061E+02
	Rank|WRST	3	6 | >	2 | <	**1** | <	4 | >	5 | >	7 | >	8 | >
	Min	9.651094E+01	2.447594E+02	**5.472274E+01**	5.472663E+01	1.289538E+02	1.452638E+02	2.816524E+02	4.566834E+02
	Rank	3	6	**1**	2	4	5	7	8
	Median	1.283495E+02	3.607668E+02	9.007684E+01	**8.258318E+01**	1.890420E+02	1.960066E+02	4.409868E+02	7.024338E+02
	Rank	3	6	2	**1**	4	5	7	8

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	(Std Dev)	(2.592926E+01)	(6.509859E+01)	(9.563140E+01)	(5.731678E+01)	(3.845185E+01)	(3.111280E+01)	(6.390306E+01)	(1.140363E+02)
f_6	Mean	1.118353E+01	3.875554E+00	6.212149E-01	**5.864706E-04**	1.499058E+00	6.806957E+00	6.219370E+00	3.205851E+01
	Rank|WRST	7	4 | <	2 | <	**1** | <	3 | <	6 | <	5 | <	8 | >
	Min	6.605830E+00	5.799000E-01	9.617000E-02	**6.000000E-05**	1.438600E-01	3.735110E+00	2.051480E+00	1.847801E+01
	Rank	7	4	2	**1**	3	6	5	8
	Median	1.135828E+01	3.181320E+00	5.360900E-01	**1.500000E-04**	1.037540E+00	6.829110E+00	6.323170E+00	3.164776E+01
	Rank	7	4	2	**1**	3	6	5	8
	(Std Dev)	(2.380259E+00)	(2.310389E+00)	(2.906478E-01)	(3.073175E-03)	(1.387208E+00)	(1.459848E+00)	(2.228597E+00)	(6.352051E+00)
f_7	Mean	3.502780E+02	5.059282E+02	3.772483E+02	3.179186E+02	**3.006720E+02**	3.976289E+02	8.013110E+02	1.129234E+03
	Rank|WRST	3	6 | >	4 | <	2 | <	**1** | <	5 | >	7 | >	8 | >
	Min	2.658569E+02	3.995895E+02	2.000127E+02	**1.792822E+02**	2.395596E+02	2.801747E+02	5.640955E+02	8.014290E+02
	Rank	4	6	2	**1**	3	5	7	8
	Median	3.503252E+02	5.018590E+02	2.988184E+02	**2.721349E+02**	2.949522E+02	4.009904E+02	8.002657E+02	1.109373E+03
	Rank	4	6	3	**1**	2	5	7	8
	(Std Dev)	(5.454765E+01)	(5.785105E+01)	(1.890138E+02)	(1.388568E+02)	(3.664387E+01)	(5.218961E+01)	(1.377843E+02)	(2.103619E+02)
f_8	Mean	1.238317E+02	3.519674E+02	1.085227E+02	**8.761533E+01**	1.822857E+02	2.006301E+02	4.607271E+02	6.392853E+02
	Rank|WRST	3	6 | >	2 | <	**1** | <	4 | >	5 | >	7 | >	8 | >
	Min	8.457144E+01	2.119256E+02	**4.577934E+01**	5.770831E+01	9.751315E+01	1.402891E+02	3.434521E+02	4.129063E+02
	Rank	3	6	**1**	2	4	5	7	8
	Median	1.203899E+02	3.502243E+02	9.452102E+01	**7.959995E+01**	1.820770E+02	1.979966E+02	4.500569E+02	6.298038E+02
	Rank	3	6	2	**1**	4	5	7	8
	(Std Dev)	(2.314138E+01)	(6.237376E+01)	(7.158344E+01)	(6.359954E+01)	(3.915689E+01)	(2.825588E+01)	(7.864919E+01)	(1.152928E+02)
f_9	Mean	7.862875E+02	9.238791E+03	1.720662E+02	**1.134185E+01**	2.191887E+02	2.983517E+03	7.858426E+03	1.878766E+04
	Rank|WRST	4	7 | >	2 | <	**1** | <	3 | <	5 | >	6 | >	8 | >
	Min	2.853134E+02	1.889633E+03	1.976621E+01	**2.582380E+00**	4.067902E+01	1.631814E+03	2.940866E+03	1.110418E+04
	Rank	4	6	2	**1**	3	5	7	8
	Median	7.608710E+02	7.520431E+03	1.438322E+02	**9.876260E+00**	1.778298E+02	2.863622E+03	7.774940E+03	1.903113E+04
	Rank	4	6	2	**1**	3	5	7	8
	(Std Dev)	(2.790664E+02)	(6.238953E+03)	(1.269757E+02)	(6.899647E+00)	(2.035450E+02)	(6.727259E+02)	(2.683522E+03)	(4.301389E+03)
f_{10}	Mean	1.157780E+04	1.302457E+04	1.458080E+04	1.270029E+04	1.379322E+04	**1.010679E+04**	1.396027E+04	1.350435E+04
	Rank|WRST	2	4 | >	8 | >	3 | =	6 | >	**1** | <	7 | >	5 | >
	Min	**7.468924E+03**	1.051264E+04	1.285161E+04	8.806641E+03	1.083847E+04	8.294871E+03	1.111100E+04	1.122080E+04
	Rank	**1**	4	8	3	5	2	6	7
	Median	1.158360E+04	1.307544E+04	1.457852E+04	1.133328E+04	1.363851E+04	**1.007173E+04**	1.381573E+04	1.343416E+04
	Rank	3	4	8	2	6	**1**	7	5
	(Std Dev)	(1.134658E+03)	(1.346860E+03)	(9.259127E+02)	(4.325205E+03)	(1.550387E+03)	(1.049051E+03)	(1.396792E+03)	(1.373541E+03)
f_{11}	Mean	9.317246E+02	**4.345439E+02**	8.634073E+02	1.072836E+03	2.860501E+03	1.080405E+03	1.407928E+03	3.032698E+03
	Rank|WRST	3	**1** | <	2 | =	4 | >	7 | >	5 | >	6 | >	8 | >
	Min	5.034358E+02	**1.619468E+02**	4.207967E+02	7.864882E+02	1.091241E+03	6.618963E+02	7.995035E+02	5.754943E+02
	Rank	3	**1**	2	6	8	5	7	4
	Median	8.991896E+02	**4.067041E+02**	8.199749E+02	1.038476E+03	2.422014E+03	1.055332E+03	1.344418E+03	2.794271E+03
	Rank	3	**1**	2	4	7	5	6	8
	(Std Dev)	(1.959595E+02)	(1.578277E+02)	(2.528406E+02)	(1.784885E+02)	(1.495907E+03)	(3.471297E+02)	(4.296599E+02)	(1.945482E+03)
f_{12}	Mean	**6.272191E+03**	2.414410E+05	2.033979E+07	8.243049E+06	1.682081E+07	1.258694E+06	8.293441E+08	9.496637E+07
	Rank|WRST	**1**	2 | >	6 | >	4 | >	5 | >	3 | >	8 | >	7 | >
	Min	**2.835171E+03**	4.793491E+04	7.453264E+05	3.493087E+06	3.774274E+05	3.472318E+05	2.408760E+07	6.107774E+06
	Rank	**1**	2	5	6	4	3	8	7
	Median	**4.143319E+03**	2.367019E+05	1.999355E+07	7.839514E+06	8.252920E+06	1.188380E+06	3.522403E+08	4.470926E+07
	Rank	**1**	2	6	4	5	3	8	7
	(Std Dev)	(7.414304E+03)	(1.000552E+05)	(1.590015E+07)	(3.202801E+06)	(2.620223E+07)	(6.032811E+05)	(1.241754E+09)	(1.623444E+08)
f_{13}	Mean	**3.677784E+02**	5.367228E+03	2.663493E+03	4.625751E+03	4.248389E+03	8.002361E+03	2.938155E+06	5.765179E+03
	Rank|WRST	**1**	5 | >	2 | >	4 | >	3 | >	7 | >	8 | >	6 | >
	Min	1.742686E+02	**1.239717E+02**	1.423638E+02	4.564860E+02	4.986277E+02	2.412066E+03	1.980229E+03	7.786561E+02
	Rank	3	**1**	2	4	5	8	7	6
	Median	**3.370455E+02**	3.003910E+03	2.528427E+03	2.807896E+03	3.093872E+03	6.273380E+03	1.168147E+04	3.619704E+03
	Rank	**1**	4	2	3	5	7	8	6
	(Std Dev)	(1.456015E+02)	(6.042762E+03)	(1.985851E+03)	(4.907714E+03)	(3.592756E+03)	(5.873221E+03)	(1.443058E+07)	(6.195785E+03)
f_{14}	Mean	**2.723554E+02**	3.949614E+04	1.287012E+06	8.006609E+05	2.769214E+05	2.379859E+05	1.129280E+05	1.764204E+06
	Rank|WRST	**1**	2 | >	7 | >	6 | >	5 | >	4 | >	3 | >	8 | >
	Min	**1.752192E+02**	4.182316E+03	1.570933E+05	3.093177E+05	2.233245E+04	9.263292E+04	1.154550E+04	1.305757E+05
	Rank	**1**	2	7	8	4	5	3	6
	Median	**2.707334E+02**	3.439578E+04	1.072499E+06	7.592860E+05	1.484396E+05	2.213742E+05	4.970062E+04	1.335026E+06
	Rank	**1**	2	7	6	4	5	3	8
	(Std Dev)	(4.837880E+01)	(2.224995E+04)	(9.711438E+05)	(3.346945E+05)	(4.395107E+05)	(9.457314E+04)	(3.301626E+05)	(1.545199E+06)
f_{15}	Mean	**1.555055E+02**	5.559175E+03	1.950291E+03	1.998950E+03	5.152163E+04	1.767691E+03	5.084433E+03	3.210780E+03
	Rank|WRST	**1**	7 | >	3 | >	4 | >	8 | >	2 | >	6 | >	5 | >
	Min	**7.785800E+01**	1.142282E+02	1.932489E+02	3.159416E+02	4.141624E+02	2.677625E+02	4.015494E+02	2.807690E+02
	Rank	**1**	2	3	6	8	4	7	5
	Median	**1.338353E+02**	3.160484E+03	9.444472E+02	1.552084E+03	1.786507E+03	8.988895E+02	1.872312E+03	2.054425E+03
	Rank	**1**	8	3	4	5	2	6	7
	(Std Dev)	(6.877350E+01)	(6.672629E+03)	(3.098894E+03)	(1.765271E+03)	(3.463227E+05)	(2.082447E+03)	(9.674096E+03)	(3.066250E+03)
f_{16}	Mean	2.715257E+03	3.458549E+03	3.527115E+03	**1.865454E+03**	3.756263E+03	3.025445E+03	3.642557E+03	4.144668E+03
	Rank|WRST	2	4 | >	5 | >	**1** | <	7 | >	3 | >	6 | >	8 | >
	Min	1.664158E+03	1.931447E+03	1.568913E+03	**5.005466E+02**	2.355655E+03	1.954742E+03	1.407238E+03	2.046524E+03
	Rank	4	5	3	**1**	8	6	2	7
	Median	2.752595E+03	3.533722E+03	3.324090E+03	**1.898265E+03**	3.840196E+03	2.994033E+03	3.703202E+03	4.066870E+03

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Rank	2	5	4	**1**	7	3	6	8
	(Std Dev)	(5.392925E+02)	(7.474259E+02)	(1.116986E+03)	(5.746903E+02)	(6.461226E+02)	(6.135331E+02)	(7.107428E+02)	(8.503244E+02)
	Mean	2.417845E+03	2.639677E+03	3.283922E+03	**1.985779E+03**	2.942427E+03	2.363026E+03	3.020686E+03	3.160173E+03
	Rank|WRST	3	4 | =	8 | >	**1** | <	5 | >	2 | =	6 | >	7 | >
	Min	1.192661E+03	1.426326E+03	1.261925E+03	**6.556463E+02**	1.289532E+03	1.082572E+03	1.674161E+03	2.173331E+03
f_{17}	Rank	3	6	4	**1**	5	2	7	8
	Median	2.453946E+03	2.638031E+03	3.496384E+03	**1.864595E+03**	2.903670E+03	2.347217E+03	3.078579E+03	3.158867E+03
	Rank	3	4	8	**1**	5	2	6	7
	(Std Dev)	(5.476858E+02)	(6.328741E+02)	(7.719910E+02)	(7.040804E+02)	(5.163397E+02)	(4.317502E+02)	(5.806584E+02)	(4.947851E+02)
	Mean	**1.795556E+02**	1.661594E+05	3.127708E+06	8.154811E+05	5.562277E+05	2.928179E+05	2.431610E+05	2.791784E+06
	Rank|WRST	**1**	2 | >	8 | >	6 | >	5 | >	4 | >	3 | >	7 | >
	Min	**1.120432E+02**	3.363045E+04	5.163363E+05	2.684693E+05	8.346638E+04	1.141744E+05	9.618162E+04	4.244896E+05
f_{18}	Rank	**1**	2	8	6	3	5	4	7
	Median	**1.737132E+02**	1.490935E+05	2.681366E+06	7.999036E+05	3.198941E+05	2.867623E+05	2.028200E+05	2.450758E+06
	Rank	**1**	2	8	6	5	4	3	7
	(Std Dev)	(3.615198E+01)	(8.235392E+04)	(1.659791E+06)	(2.441233E+05)	(7.870493E+05)	(8.181366E+04)	(1.525892E+05)	(1.953294E+06)
	Mean	**1.915620E+02**	6.138026E+03	2.045224E+03	1.784556E+03	2.721021E+03	1.855218E+03	3.996112E+04	3.971079E+03
	Rank|WRST	**1**	7 | >	4 | >	2 | >	5 | >	3 | >	8 | >	6 | >
	Min	8.204230E+01	9.533120E+01	**5.190170E+01**	1.988803E+02	1.935004E+02	2.710994E+02	1.415717E+02	1.348286E+02
f_{19}	Rank	2	3	**1**	7	6	8	5	4
	Median	**1.482601E+02**	2.686237E+03	4.851651E+02	1.174854E+03	1.657180E+03	9.741167E+02	1.088136E+03	1.519876E+03
	Rank	**1**	8	2	5	7	3	4	6
	(Std Dev)	(1.489906E+02)	(8.044927E+03)	(3.307473E+03)	(1.714498E+03)	(3.040036E+03)	(2.258100E+03)	(2.713754E+05)	(4.825721E+03)
	Mean	2.087782E+03	2.690199E+03	3.014567E+03	**1.380418E+03**	2.406939E+03	1.886587E+03	2.604035E+03	2.951311E+03
	Rank|WRST	3	6 | >	8 | >	**1** | <	4 | >	2 | <	5 | >	7 | >
	Min	1.299881E+03	1.421215E+03	9.598696E+02	**4.383888E+02**	1.541075E+03	1.137621E+03	1.300961E+03	1.659263E+03
f_{20}	Rank	4	6	2	**1**	7	3	5	8
	Median	2.030397E+03	2.664662E+03	3.054368E+03	**1.377838E+03**	2.360794E+03	1.853459E+03	2.637041E+03	2.952433E+03
	Rank	3	6	8	**1**	4	2	5	7
	(Std Dev)	(4.355347E+02)	(6.267003E+02)	(5.840032E+02)	(4.325510E+02)	(4.441410E+02)	(4.601056E+02)	(5.374602E+02)	(5.096689E+02)
	Mean	3.691882E+02	5.725085E+02	3.522737E+02	**3.291080E+02**	4.380857E+02	4.505221E+02	7.136165E+02	8.523056E+02
	Rank|WRST	3	6 | >	2 | <	**1** | <	4 | >	5 | >	7 | >	8 | >
	Min	3.088941E+02	4.640510E+02	2.870720E+02	**2.868987E+02**	3.842810E+02	3.797721E+02	5.250441E+02	6.707442E+02
f_{21}	Rank	3	6	2	**1**	5	4	7	8
	Median	3.619496E+02	5.738673E+02	3.385765E+02	**3.282266E+02**	4.418423E+02	4.462910E+02	6.944656E+02	8.459071E+02
	Rank	3	6	2	**1**	4	5	7	8
	(Std Dev)	(3.118132E+01)	(6.311894E+01)	(8.056047E+01)	(1.776872E+01)	(2.927858E+01)	(3.151877E+01)	(1.177440E+02)	(8.387623E+01)
	Mean	1.094758E+04	1.446545E+04	1.563347E+04	1.208544E+04	1.448261E+04	**1.030725E+04**	1.506461E+04	1.615558E+04
	Rank|WRST	2	4 | >	7 | >	3 | <	5 | >	**1** | =	6 | >	8 | >
	Min	**1.000000E+02**	1.113181E+04	9.784559E+03	9.155325E+03	1.151168E+04	**1.000000E+02**	1.511609E+02	**1.000000E+02**
f_{22}	Rank	**1**	7	6	5	8	**1**	4	**1**
	Median	1.247748E+04	1.460267E+04	1.576480E+04	**1.147068E+04**	1.456918E+04	1.225359E+04	1.596861E+04	1.765104E+04
	Rank	3	5	6	**1**	4	2	7	8
	(Std Dev)	(4.456681E+03)	(1.442361E+03)	(1.675188E+03)	(2.614243E+03)	(1.365287E+03)	(4.918313E+03)	(3.988345E+03)	(4.460663E+03)
	Mean	7.163644E+02	8.395089E+02	**6.547822E+02**	8.072490E+02	8.190664E+02	9.482705E+02	1.100021E+03	1.260377E+03
	Rank|WRST	2	5 | >	**1** | <	3 | >	4 | >	6 | >	7 | >	8 | >
	Min	6.283171E+02	6.992947E+02	**6.147076E+02**	6.524893E+02	7.203130E+02	8.085698E+02	8.008422E+02	9.844235E+02
f_{23}	Rank	2	4	**1**	3	5	7	6	8
	Median	6.880948E+02	8.432052E+02	**6.522762E+02**	7.964283E+02	8.195646E+02	9.434872E+02	1.071229E+03	1.261715E+03
	Rank	2	5	**1**	3	4	6	7	8
	(Std Dev)	(8.802764E+01)	(6.015380E+01)	(2.560236E+01)	(8.945664E+01)	(4.892824E+01)	(6.424586E+01)	(1.750807E+02)	(1.360133E+02)
	Mean	**1.009914E+03**	1.212613E+03	1.037547E+03	1.233356E+03	1.196236E+03	1.298970E+03	1.456328E+03	2.114579E+03
	Rank|WRST	**1**	4 | >	2 | >	5 | >	3 | >	6 | >	7 | >	8 | >
	Min	**9.414849E+02**	1.109777E+03	9.807116E+02	1.028489E+03	1.071493E+03	1.144987E+03	1.154604E+03	1.720931E+03
f_{24}	Rank	**1**	5	2	3	4	6	7	8
	Median	**9.925132E+02**	1.201879E+03	1.032220E+03	1.195066E+03	1.194267E+03	1.291370E+03	1.429685E+03	2.057118E+03
	Rank	**1**	5	2	4	3	6	7	8
	(Std Dev)	(4.928013E+01)	(5.432552E+01)	(3.151828E+01)	(1.356058E+02)	(6.654372E+01)	(9.592666E+01)	(1.806374E+02)	(2.930625E+02)
	Mean	**6.758304E+02**	7.486147E+02	1.070835E+03	7.562412E+02	1.232055E+03	8.247413E+02	1.650464E+03	1.474689E+03
	Rank|WRST	**1**	2 | >	5 | >	3 | >	6 | >	4 | >	8 | >	7 | >
	Min	**5.770996E+02**	6.116229E+02	9.705492E+02	6.665727E+02	9.978727E+02	6.816388E+02	1.112002E+03	1.037213E+03
f_{25}	Rank	**1**	2	5	3	6	4	8	7
	Median	**6.983352E+02**	7.450242E+02	1.070288E+03	7.625645E+02	1.218462E+03	8.335116E+02	1.661978E+03	1.425337E+03
	Rank	**1**	2	5	3	6	4	8	7
	(Std Dev)	(4.775175E+01)	(6.124668E+01)	(4.810690E+01)	(3.895707E+01)	(1.348333E+02)	(5.844039E+01)	(2.728716E+02)	(2.700841E+02)
	Mean	**2.542470E+03**	6.833489E+03	4.518602E+03	4.119213E+03	5.574612E+03	7.351520E+03	1.500473E+04	1.438311E+04
	Rank|WRST	**1**	5 | >	3 | >	2 | >	4 | >	6 | >	8 | >	7 | >
	Min	**3.000000E+02**	5.334118E+03	3.789577E+03	3.422258E+03	4.236304E+03	5.536617E+03	1.933274E+03	3.010774E+02
f_{26}	Rank	**1**	7	5	4	6	8	3	2
	Median	**3.000001E+02**	6.771797E+03	4.486363E+03	4.124809E+03	5.442238E+03	7.337922E+03	1.579722E+04	1.449041E+04
	Rank	**1**	5	3	2	4	6	8	7
	(Std Dev)	(2.942707E+03)	(8.016257E+02)	(3.684093E+02)	(3.510567E+02)	(5.566002E+02)	(8.231027E+02)	(4.301092E+03)	(4.188147E+03)
	Mean	7.513341E+02	7.454203E+02	7.598011E+02	**5.162884E+02**	1.111119E+03	9.424340E+02	1.022088E+03	1.642000E+03
	Rank|WRST	3	2 | =	4 | =	**1** | <	7 | >	5 | >	6 | >	8 | >
	Min	6.467411E+02	6.418840E+02	6.862727E+02	**5.000234E+02**	8.798638E+02	7.910390E+02	8.151996E+02	1.120755E+03
f_{27}	Rank	3	2	4	**1**	7	5	6	8

f_n	Criteria	IMPSOwithCMAR	EAPSO	AMSEPSO	EOPSO	PSO-sono	PPSO	MPSO	TCSPSO
	Median	7.400162E+02	7.372442E+02	7.612646E+02	**5.000238E+02**	1.089967E+03	9.452293E+02	9.750889E+02	1.682112E+03
	Rank	3	2	4	**1**	7	5	6	8
	(Std Dev)	(6.434876E+01)	(5.076946E+01)	(3.643868E+01)	(4.958592E+01)	(1.356630E+02)	(8.253182E+01)	(1.564907E+02)	(3.248607E+02)
f_{28}	Mean	**4.405444E+02**	5.276895E+02	8.026106E+02	5.565729E+02	1.155916E+03	5.903851E+02	2.606181E+03	1.880288E+03
	Rank|WRST	**1**	2 | >	5 | >	3 | >	6 | >	4 | >	8 | >	7 | >
	Min	**3.000000E+02**	4.635943E+02	7.055741E+02	5.128589E+02	8.357839E+02	4.913092E+02	9.165695E+02	9.142338E+02
	Rank	**1**	2	5	4	6	3	8	7
	Median	**4.965012E+02**	5.216694E+02	7.972706E+02	5.573908E+02	1.125497E+03	5.832563E+02	2.634631E+03	1.728019E+03
	Rank	**1**	2	5	3	6	4	8	7
	(Std Dev)	(1.063049E+02)	(3.837218E+01)	(5.515244E+01)	(2.917531E+01)	(1.405203E+02)	(3.648989E+01)	(8.542891E+02)	(7.590176E+02)
f_{29}	Mean	3.138869E+03	2.947696E+03	2.884844E+03	**2.298771E+03**	3.408312E+03	3.420070E+03	3.516463E+03	4.361319E+03
	Rank|WRST	4	3 | =	2 | <	**1** | <	5 | >	6 | >	7 | >	8 | >
	Min	1.811730E+03	1.951433E+03	1.770830E+03	**8.787588E+02**	1.891635E+03	1.926388E+03	2.241999E+03	2.689789E+03
	Rank	3	6	2	**1**	4	5	7	8
	Median	3.047393E+03	2.915526E+03	2.686001E+03	**2.033305E+03**	3.536227E+03	3.314266E+03	3.470005E+03	4.387856E+03
	Rank	4	3	2	**1**	7	5	6	8
	(Std Dev)	(5.986308E+02)	(5.452312E+02)	(7.701929E+02)	(1.122032E+03)	(7.032335E+02)	(5.016035E+02)	(5.862053E+02)	(7.181784E+02)
f_{30}	Mean	6.057618E+03	6.706293E+03	1.914339E+04	**4.073641E+03**	2.114684E+05	1.071450E+04	8.198312E+06	6.392487E+05
	Rank|WRST	2	3 | =	5 | >	**1** | <	6 | >	4 | >	8 | >	7 | >
	Min	3.560103E+03	2.924057E+03	7.565690E+03	**3.259723E+02**	7.924672E+03	4.516134E+03	4.493885E+03	3.782977E+04
	Rank	3	2	6	**1**	7	5	4	8
	Median	5.518555E+03	5.108956E+03	1.614366E+04	**3.221791E+03**	6.909379E+04	9.229868E+03	2.652425E+05	2.711676E+05
	Rank	3	2	5	**1**	6	4	7	8
	(Std Dev)	(1.700558E+03)	(3.676308E+03)	(1.182104E+04)	(3.395733E+03)	(2.941233E+05)	(4.992505E+03)	(2.323847E+07)	(1.549375E+06)

References

[1] R. Mallipeddi, S. Mallipeddi, P. Suganthan, Ensemble strategies with adaptive evolutionary programming, Information Sciences 180 (9) (2010) 1571–1581.

[2] S. M. Elsayed, R. A. Sarker, D. L. Essam, N. M. Hamza, Testing united multi-operator evolutionary algorithms on the cec2014 real-parameter numerical optimization, in: 2014 IEEE Congress on Evolutionary Computation, 2014, pp. 1650–1657.

[3] S. Elsayed, N. Hamza, R. Sarker, Testing united multi-operator evolutionary algorithms-ii on single objective optimization problems, in: 2016 IEEE Congress on Evolutionary Computation, 2016, pp. 2966–2973.

[4] A. Kumar, R. K. Misra, D. Singh, Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase, in: 2017 IEEE Congress on Evolutionary Computation, 2017, pp. 1835–1842.

[5] N. Lynn, P. N. Suganthan, Ensemble particle swarm optimizer, Applied Soft Computing 55 (2017) 533–548.

[6] G. Wu, R. Mallipeddi, P. N. Suganthan, Ensemble strategies for population-based optimization algorithms–a survey, Swarm and evolutionary computation 44 (2019) 695–711.

[7] W. Yi, Y. Chen, Z. Pei, J. Lu, Adaptive differential evolution with ensembling operators for continuous optimization problems, Swarm and Evolutionary Computation 69 (2022) 100994.

[8] Z. Tan, Y. Tang, K. Li, H. Huang, S. Luo, Differential evolution with hybrid parameters and mutation strategies based on reinforcement learning, Swarm and Evolutionary Computation 75 (2022) 101194.

[9] L. Hong, Y. Guo, F. Liu, B. Wang, A variant of the united multi-operator evolutionary algorithms using sequential quadratic programming and improved shade-cnepsin, Information Sciences 622 (2023) 652–681.

[10] L. Hong, X. Yu, B. Wang, J. Woodward, E. Özcan, An improved ensemble particle swarm optimizer using niching behavior and covariance matrix adapted retreat phase, Swarm and Evolutionary Computation 78 (2023) 101278.

[11] V. Stanovov, S. Akhmedova, E. Semenkin, Lshade algorithm with rank-based selective pressure strategy for solving cec 2017 benchmark problems, in: 2018 IEEE Congress on Evolutionary Computation, 2018, pp. 1–8.

[12] J. Yang, J. Yu, C. Huang, Adaptive multistrategy ensemble particle swarm optimization with signal-to-noise ratio distance metric, Information Sciences 612 (2022) 1066–1094.

[13] H. T. Kahraman, S. Aras, E. Gedikli, Fitness-distance balance (fdb): A new selection method for meta-heuristic search algorithms, Knowledge-Based Systems 190 (2020) 105169.

[14] N. H. Awad, M. Z. P. Ali, N. Suganthan, J. J. Liang, B. Y. Qu, Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization, Nanyang Technological University, Singapore and Jordan University of Science and Technology, Jordan and Zhengzhou University, Zhengzhou, China, Technical Report (2017).

[15] F. E. F. Junior, G. G. Yen, Particle swarm optimization of deep neural networks architectures for image classification, Swarm and Evolutionary Computation 49 (2019) 62–74.

[16] A. Darwish, D. Ezzat, A. E. Hassanien, An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis, Swarm and evolutionary computation 52 (2020) 100616.

[17] A. Dixit, A. Mani, R. Bansal, Cov2-detect-net: Design of covid-19 prediction model based on hybrid de-pso with svm using chest x-ray images, Information Sciences 571 (2021) 676–692.

[18] B. Bai, Z. Guo, C. Zhou, W. Zhang, J. Zhang, Application of adaptive reliability importance sampling-based extended domain pso on single mode failure in reliability engineering, Information Sciences 546 (2021) 42–59.

[19] A. Ben Ali, G. Luque, E. Alba, An efficient discrete pso coupled with a fast local search heuristic for the dna fragment assembly problem, Information Sciences 512 (2020) 880–908.

[20] S. Ding, T. Zhang, C. Chen, Y. Lv, B. Xin, Z. Yuan, R. Wang, P. M. Pardalos, An efficient particle swarm optimization with evolutionary multitasking for stochastic area coverage of heterogeneous sensors, Information Sciences 645 (2023) 119319.

[21] A.-D. Li, B. Xue, M. Zhang, Multi-objective particle swarm optimization for key quality feature selection in complex manufacturing processes, Information Sciences 641 (2023) 119062.

[22] X. Yang, Y. Qi, H. Chen, B. Liu, W. Liu, Generation-based parallel particle swarm optimization for adversarial text attacks, Information Sciences 644 (2023) 119237.

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

[24] H. Liu, X.-W. Zhang, L.-P. Tu, A modified particle swarm optimization using adaptive strategy, Expert systems with applications 152 (2020) 113353.

[25] T. Li, J. Shi, W. Deng, Z. Hu, Pyramid particle swarm optimization with novel strategies of competition and cooperation, Applied Soft Computing 121 (2022) 108731.

[26] Z. Meng, Y. Zhong, G. Mao, Y. Liang, Pso-sono: A novel pso variant for single-objective numerical optimization, Information Sciences 586 (2022) 176–191.

[27] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360), IEEE, 1998, pp. 69–73.

[28] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Information Sciences 291 (2015) 43–60.

[29] S. Zhao, D. Wang, Elite-ordinary synergistic particle swarm optimization, Information Sciences 609 (2022) 1567–1587.

[30] Y. Zhang, Elite archives-driven particle swarm optimization for large scale numerical optimization and its engineering applications, Swarm and Evolutionary Computation 76 (2023) 101212.

[31] N. Hansen, S. D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es), Evolutionary computation 11 (1) (2003) 1–18.

[32] N. H. Awad, M. Z. Ali, P. N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation, 2017, pp. 372–379.

[33] A. Kumar, R. K. Misra, D. Singh, Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase, in: 2017 IEEE Congress on Evolutionary Computation, 2017, pp. 1835–1842.

[34] N. Hansen, The cma evolution strategy: A tutorial, arXiv preprint arXiv:1604.00772 (2016).

[35] H. Modares, M.-B. Naghibi Sistani, Solving nonlinear optimal control problems using a hybrid ipso–sqp algorithm, Engineering Applications of Artificial Intelligence 24 (3) (2011) 476–484.

[36] Y. Zhang, F. Yao, H. H.-C. Iu, T. Fernando, K. P. Wong, Sequential quadratic programming particle swarm optimization for wind power system operations considering emissions, Journal of Modern Power Systems and Clean Energy 1 (3) (2013) 231–240.

[37] C. B. Costa, A. C. da Costa, R. M. Filho, Mathematical modeling and optimal control strategy development for an adipic acid crystallization process, Chemical Engineering and Processing: Process Intensification 44 (7) (2005) 737–753.

[38] J. Derrac, S. García, D. Molina, F. Herrera, 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 (1) (2011) 3–18.