Complex & Intelligent Systems (2024) 10:2421–2443
https://doi.org/10.1007/s40747-023-01269-z

ORIGINAL ARTICLE

A sequential quadratic programming based strategy for particle swarm optimization on single-objective numerical optimization

Libin Hong[1] · Xinmeng Yu[1] · Guofang Tao[1] · Ender Özcan[2] · John Woodward[3]

Received: 5 April 2023 / Accepted: 12 October 2023 / Published online: 22 November 2023

Abstract
Over the last decade, particle swarm optimization has become increasingly sophisticated because well-balanced exploration and exploitation mechanisms have been proposed. The sequential quadratic programming method, which is widely used for real-parameter optimization problems, demonstrates its outstanding local search capability. In this study, two mechanisms are proposed and integrated into particle swarm optimization for single-objective numerical optimization. A novel ratio adaptation scheme is utilized for calculating the proportion of subpopulations and intermittently invoking the sequential quadratic programming for local search start from the best particle to seek a better solution. The novel particle swarm optimization variant was validated on CEC2013, CEC2014, and CEC2017 benchmark functions. The experimental results demonstrate impressive performance compared with the state-of-the-art particle swarm optimization-based algorithms. Furthermore, the results also illustrate the effectiveness of the two mechanisms when cooperating to achieve significant improvement.

Keywords Particle swarm optimization · Ratio adaptation scheme · Sequential quadratic programming · Single-objective numerical optimization

Introduction

Particle swarm optimization (PSO) is a well-known population-based metaheuristic algorithm [1]. Many PSO-based variants and their applications were proposed in the last decade [2–12]. PSO applied for large-scale group decision-making [13], adsorption control of pipeline robot [14], health estimation for electric vehicle [15], substitution box construction based on quantum-inspired quantum walks [16], feature-related time consumption reduction for surface electromyography [17], and credit risk assessment for personal auto loan [18]. Li et al. [19] indicated that the research directions of the PSO algorithm can be divided into four categories: parameter tuning, topology choices, learning strategy improvements, and integration with other algorithms. Among the large number of PSO-based investigations, hybridization is a popular approach because it can combine several complementary PSO variants or algorithms into a solid framework. In [20], a hybrid genetic algorithm with PSO for multimodal functions was proposed. In [21], search mechanisms of swallow swarm optimization (SSO) were implemented with PSO to formulate the hybrid particle swallow swarm optimization (HPSSO) algorithm. In [22] a hybrid feature selection algorithm based on PSO was proposed; this variant uses a local search strategy, which is embedded in the PSO, to select the less correlated and salient feature subset. In [23], a hybrid PSO algorithm that utilizes an adaptive learning strategy (ALPSO) was proposed; a self-learning-based candidate generation strategy and a competitive learning-based prediction strategy were employed to ensure the exploration ability and the exploitation of the algorithm, respectively. In [24], the variable neighborhood search (VNS) was used to solve the clustered vehicle routing problem (CluVRP) was employed for the PSO to ensure solution intensity and bring the solution to the local optima. In [25], the inertia weight PSO (iwPSO) [26] and social learning PSO (SLPSO) [27] were combined to form a PSO on single-objective numerical optimization (PSO-sono). The PSO-sono approach demon-

✉ Libin Hong
libin.hong@hznu.edu.cn

1 School of Information Science and Technology, Hangzhou Normal University, No. 2318 Yuhangtang Road, Hangzhou 31121, People's Republic of China

2 School of Computer Science, University of Nottingham, Wollaton Road, Nottingham NG8 1BB, UK

3 Department of Computer Science, Loughborough University, Epinal Way, Loughborough LE11 3TU, UK

strates outstanding performance on CEC2013, CEC2014, and CEC2017 benchmark functions.

The essence of hybridization is to select complementary algorithms and exert their respective strengths; however, unsuitable subpopulation size may limit the strengths. According to our observation, subpopulation allocation considerably influences the algorithm performance due to the economy of scale in diverse stages of evolution. Furthermore, the exploitation capability increases the convergence rate, especially of particles near the global optimum. Hence, employing a proper local search method can improve evolution efficiency. The sequential quadratic programming (SQP) method was proposed by Wilson [28] due to its outstanding local search capability, and it is widely used in real-parameter optimization problems [29–33]. The SQP method can be used to seek the local minimum satisfying the constraints. Theories related to the SQP method can be found in [28, 34–36].

In this study, the PSO-sono was further reformed using a novel ratio adaptation scheme (NRAS) for calculating the proportion of subpopulations and applying the SQP method intermittently to the best particle generated either by iwPSO or SLPSO, illustrating impressive performance. The proposed PSO-based variant is called SQPPSO-sono, and it performs more competitively than the state-of-the-art algorithms: PSO-sono [25], ensemble particle swarm optimizer (EPSO) [37], pyramid particle swarm optimization (PPSO) [38], modified particle swarm optimization (MPSO) [39], terminal crossover and steering-based particle swarm optimization (TCSPSO) [40], and heterogeneous comprehensive learning particle swarm optimization (HCLPSO) [41]. All the compared algorithms were tested on CEC2013, CEC2014, and CEC2017 benchmark functions. The experimental results demonstrate that SQPPSO-sono not only has outstanding performance in most cases, but also better local search capability when it cooperates with NRAS. The main highlights of this work are the following:

- *The SQP method and NRAS are introduced in the SQPPSO-sono approach, and the two mechanisms cooperate and significantly promote enhanced performance.*
- *The SQP method can result in a sharp convergence rate in the early optimization process due to application on the current best particle. It enhances the local search capability of SQPPSO-sono and promotes rankings compared with other algorithms.*
- *NRAS is proposed and inspired by an effective butterfly optimizer using the covariance matrix adapted retreat phase* [36], *for dividing subpopulations.*
- *A large test suite including all the benchmark functions from CEC2013, CEC2014, and CEC2017 test suites is used for algorithm validation. To some extent, the overfitting problem can be avoided owing to the large number of tests for real-parameter single-objective optimization compared with employing only one test suite including a few benchmark functions.*

The remainder of this paper is organized as follows. "Particle swarm optimization and PSO-sono" represents the PSO and PSO-sono. "The sequential quadratic programming based particle swarm optimization" represents the novel strategy and the proposed SQPPSO-sono algorithm. The integration and implementation of NRAS and the SQP method are described in detail. "Performance evaluation" presents the experimental results and the parameter settings of the compared algorithms. Comparisons with the recent state-of-the-art PSO-based algorithms, analysis, discussions, and future research directions are presented in this section as well. "Conclusions" concludes and summarizes the paper.

Particle swarm optimization and PSO-sono

PSO-sono is a variant of PSO, with the following improvements: (1) A sorted particle swarm with hybrid paradigms improves the optimization performance; (2) Novel adaptation schemes for the ratio of each paradigm and constriction coefficients during evolution; (3) A fully-informed search scheme based on the current best particle in each generation assists PSO to jump out of the local optimum and improve performance.

Particle swarm optimization

The velocity and position, which are the basic properties of the particles of the classical PSO, are updated for each particle [42]. The velocity and position of the ith particle at generation t are recorded as $V_i(t)$ and $X_i(t)$, respectively, and the formulas can be presented as follows:

$$V_i^d(t+1) = V_i^d(t) + c_1 * r1_i^d * (P\text{best}_i^d(t) - X_i^d(t)) + c_2 * r2_i^d * (G\text{best}^d(t) - X_i^d(t)) \quad (1)$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \quad (2)$$

$r1_i^d$ and $r2_i^d$ are randomly generated by uniform distribution in range [0,1], $d = 1, 2, \ldots, \text{dim}$, where dim is the dimension size. $P\text{best}_i^d(t)$ is the ith particle's previous best solution, and $G\text{best}(t)^d$ is the whole swarm's best solution, which are as defined below:

$$P\text{best}_i^d(t) = \min\{\text{fit}(X_i^d(1)), \text{fit}(X_i^d(2)), \ldots\ldots, \text{fit}(X_i^d(t))\} \quad (3)$$

$$G\text{best}^d(t) = \min\{\text{fit}(P\text{best}_1^d(t)), \text{fit}(P\text{best}_2^d(t)), \ldots\ldots, \text{fit}(P\text{best}_N^d(t))\} \quad (4)$$

Inertia weight PSO and social learning PSO

Meng et al. [25] proposed to combine two complementary PSO variants, iwPSO and SLPSO, to promote the performance of single-objective numerical optimization, named PSO-sono. The iwPSO algorithm uses following equations to calculate the particle's velocity and position:

$$V_i^d(t+1) = w * V_i^d(t) + c_1 * r1_i^d * (P\text{best}_i^d(t) - X_i^d(t)) + c_2 * r2_i^d * (G\text{best}^d(t) - X_i^d(t)) \tag{5}$$

where w is the inertia weight of the velocity. $r1_i^d$ and $r2_i^d$ are randomly generated by uniform distribution in range [0,1], $d = 1, 2, ..., \text{dim}$. $P\text{best}_i^d(t)$ is the ith particle's previous best experience, and $G\text{best}(t)^d$ is the whole swarm's best experience.

The SLPSO algorithm uses Eqs. 6 and 7 to update the particle's velocity and position:

$$X_i^d(t+1) = \begin{cases} X_i^d(t) + V_i^d(t+1), & \text{if} \quad p_i(t) \le lp_i \\ X_i^d(t), & \text{otherwise.} \end{cases} \tag{6}$$

where $V_i^d(t+1)$ satisfies:

$$\begin{cases} I_i^d(t) = X_k^d(t) - X_i^d(t) \\ C_i^d(t) = \overline{X}_{\text{center}}^d(t) - X_i^d(t) \\ V_i^d(t+1) = r1_i * V_i^d(t) + r2_i^d * I_i^d(t) + r3_i^d * \epsilon * C_i^d(t) \end{cases} \tag{7}$$

where $X_k^d(t)$ is a randomly selected better particle of the ith particle with dth dimension in the tth generation, the ith particle is automatically selected if it is the best particle. $\overline{X}_{\text{center}}^d(t)$ is the center of the population. ϵ is the social influence, $\epsilon = 0.01 * \frac{\text{dim}}{100}$. $r0_i^d$, $r1_i^d$, and $r2_i^d$ are randomly generated by uniform distribution in range [0,1], $d = 1, 2, ..., \text{dim}$. lp_i denotes the learning probability of the ith particle, and it obeys the following:

$$lp_i = \left(1 - \frac{i-1}{ps}\right)^{ln(\lceil \frac{\text{dim}}{100} \rceil)} \tag{8}$$

The sequential quadratic programming based particle swarm optimization

Compared with the PSO-sono, the proposed SQPPSO-sono makes two significant improvements. The improvements in the red boxes are shown in Fig. 1. The framework of SQPPSO-sono are given in Algorithm 1. Figure 1 indicates that the NRAS and SQP methods are embedded into the SQPPSO-sono algorithm at the beginning of the generation and after the best particle is produced by iwPSO or SLPSO. The NRAS method is applied after the initialization and calculates the ratio r as a proportion of the subpopulation at the beginning of each generation. Then, iwPSO and SLPSO are run simultaneously according to the allocations, and a global best particle is generated from both subpopulations. Later, the SQP method is applied to further enhance the exploitation capability by starting from the current global best particle. When the termination condition is satisfied, the algorithm stops running. The major contributions of the proposed algorithm are summarized below:

- *A novel ratio adaptation scheme is employed to calculate the proportion of each subpopulation for iwPSO and SLPSO rather than using the numbers of success particles to calculate the proportion of subpopulations.*
- *The SQP method is employed to execute local search starting from the current global best particle, either generated by iwPSO or SLPSO, to search a better solution in the current generation. If a better solution is found, then the best particle is replaced by the better solution found by the SQP method.*

Novel ratio adaptation scheme

In PSO-sono, by sorting the particles based on their fitness values, the population of particles is divided into two groups: the better-particle-group ($PS1$) and the worse-particle-group ($PS2$), which are evolved by iwPSO and SLPSO, respectively. The ratio r is used for calculating the proportion of the better particles in the entire population; thus, the proportion of the worse particles is $1 - r$. The ratio r in PSO-sono is represented as follows:

$$r = \frac{ns_b}{ns_b + ns_w}, \tag{9}$$

where ns_b and ns_w represent the amount of success particles in the better-particle-group and worse-particle-group, respectively. Furthermore, a truncation readjustment of r is involved when its value is in range [0.1, 0.9].

In SQPPSO-sono, a novel ratio adaptation scheme (NRAS) is proposed, which is inspired by [36]. An adaptation scheme of ratio is used for selecting strategy in [36], while here NRAS is used for dividing subpopulations. Two factors are considered in calculating the ratio r:

- *The quality of particles: the lower fitness value, the higher quality of particle obtained.*
- *The diversity rate of particles in each subpopulation.*

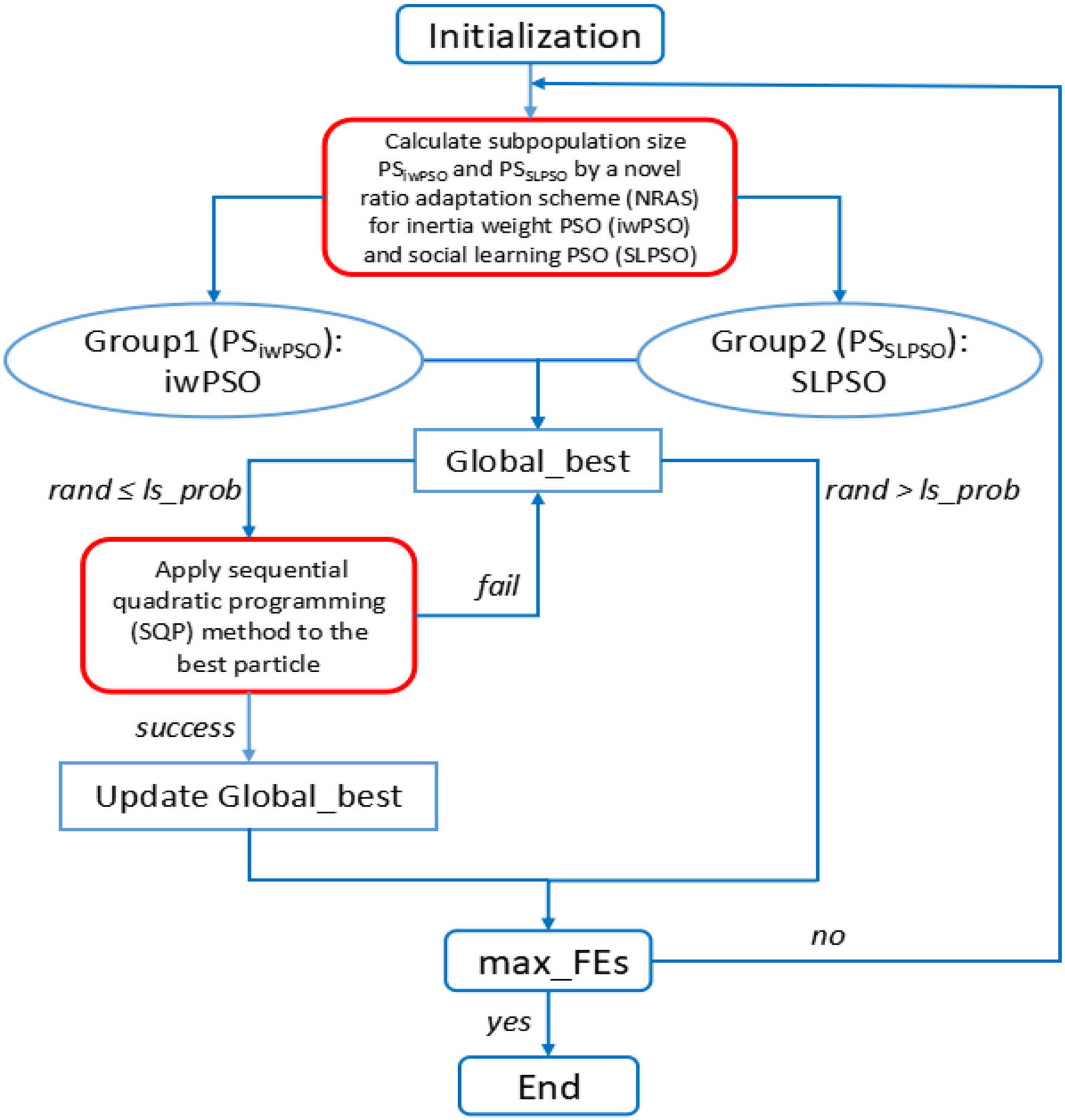

Fig. 1 Main workflow of SQPPSO-sono

The normalized quality values (nq) are calculated as follows:

$$nq_i = \frac{\text{fitness}(X_{PS_i,\text{best}})}{\text{fitness}(X_{PS_1,\text{best}}) + \text{fitness}(X_{PS_2,\text{best}})}, \ \forall i = 1, 2. \tag{10}$$

The diversity rate is calculated as follows:

$$\text{div}_i = \sum_{z=2}^{PS_i} \text{dis}(X_{PS_i,z}, X_{PS_i,\text{best}}), \ \forall i = 1, 2, \tag{11}$$

where $\text{dis}(X_{PS_i,z}, X_{PS_i,\text{best}})$ denotes the Euclidean distance between the zth particle and best particle in PS_i. The normalized diversity (nd) is calculated as follows:

$$nd_i = \frac{\text{div}_i}{\text{div}_1 + \text{div}_2}, \ \forall i = 1, 2. \tag{12}$$

In Eq. 9, only the fitness values of particles were considered to adjust the subpopulation size for PSO-sono. This algorithm may fall easily into a local optimum. However, the proposed NRAS considers both fitness value and diversity of particles.

$$k_i = (1 - nq_i) + nd_i, \forall i = 1, 2 \tag{13}$$

$$r = \max\left(0.1, \min\left(0.9, \frac{k_1}{k_1 + k_2}\right)\right). \tag{14}$$

Strategy using sequential quadratic programming

In [25], PSO-sono applies the fully-informed search scheme on the best particle in each generation to help the algorithm jump out the local optimum. In contrast, the proposed SQPPSO-sono algorithm replaces the fully-informed search scheme with the SQP method to enhance the local search capability. Due to function evaluations (FEs) are performed in the SQP method, the function evaluations of the SQP method are included in FEs to achieve a fair comparison.

The essence of the SQP method is to transform a nonlinear problem into a linear problem [43]?. The principle of the SQP method is to find a decent direction and models a quadratic optimal problem. The nonlinear optimization problems can be represented as follows:

$$\text{minimize } f(x), \text{ such that } \begin{cases} h(x) = 0 \\ g(x) \le 0 \end{cases} \tag{15}$$

Algorithm 1 General framework of SQPPSO-sono

Define $PS = PS_{iwPSO} + PS_{SLPSO}$, $ratio = 0.5$, $Prob_{ls} = 0.1$, $T = 0$, $G = 1$ and set parameters.
for i = 1 : PS **do**
  Initialize the ith particle, $X_{i,G}$.
end for
while termination condition is not satisfied **do**
  $T = T + 1$
  if $T == \frac{PS}{2}$ **then**
    Update $ratio$ by, the novel ratio adaptation scheme (NRAS), Equations 13 and 14.
    $T \leftarrow 0$
  end if
  Sort the population in descending order.
  Separate the sorted population into two groups according to the $ratio$.
  Update particles in the 1st group using Equations 2 and 5.
  Update particles in the 2nd group using Equations 2 and 7.
  if rand(0,1) $\le Prob_{ls}$ **then**
    Apply sequential quadratic programming (SQP) to the best particle to find a solution.
    if the solution is improved **then**
      $Prob_{ls} = 0.1$
      Update the best particle.
    else
      $Prob_{ls} = 0.01$
    end if
  end if
  $G = G + 1$
end while

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

$$L(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x), \tag{16}$$

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

$$\text{minimize } f^{'}(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T Hf(x_k)(x - x_k), \quad \text{such that } \begin{cases} h(x_k) + h^{'}(x_k)(x - x_k) = 0 \\ g(x_k) + g^{'}(x_k)(x - x_k) \le 0 \end{cases} \tag{17}$$

Here, $Hf(x_k)$ is the Hessian of f at $x \in R^n$. Because the SQP method highly relies on the initial estimate [44], the proposed SQPPSO-sono only applies the SQP method to the current global best particle. Thus, the method is used to enhance the local search capability for the proposed algorithm.

Performance evaluation

The recent PSO-based variants were selected in this study to evaluate the performance of SQPPSO-sono, and tests based on the CEC2013, CEC2014, and CEC2017 benchmark functions were performed. The PSO-based variants, namely PSO-sono, EPSO, PPSO, MPSO, TCSPSO, and HCLPSO, were considered. For the sake of fairness, the FEs for each variant with the same dimension size were set to identical values.

Parameter settings

The general form of maximum FEs is $10^4 \times$ D; hence, FEs for all tested algorithms were set to 10×10^4, 30×10^4, 50×10^4, and 10×10^5 for 10D, 30D, 50D, and 100D, respectively. Table 1 lists the parameter settings for the proposed SQPPSO-sono algorithm and the compared PSO variants. We used the recommended default settings, tuned by original investigators and yielding the best performance for the state-of-the-art PSO variants.

Experimental results

To better measure the performance and reliability of the proposed algorithm, the statistics of rankings for SQPPSO-sono, PSO-sono, MPSO, PPSO, MPSO, TCSPSO, and HCLPSO, which are tested on CEC2013, CEC2014, and CEC2017 benchmark functions, are presented in Tables 2, 3, and 4, respectively. The top 3 (1st, 2nd, and 3rd) rankings of the mean, minimum, and median values are counted for the 10, 30, and 50-dimensional tests for all compared variants on CEC2013 and CEC2014, while the statistics of 10, 30, 50,

Table 1 Parameter settings for the compared PSO variants

Algorithm	Default parameter settings
SQPPSO-sono	Pop size = 100 (for 10, 30, 50 and 100D), w = 0.9 → 0.4, $\epsilon = \frac{\dim}{PS} * 0.01$
PSO-sono	Pop size = 100 (for 10, 30, 50 and 100D), w = 0.9 → 0.4, $\epsilon = \frac{\dim}{PS} * 0.01$, r = 0.5
EPSO	Pop size for 10D = 20 (8/12), pop size for 30, 50 and 100D = 40 (15/25), default settings of its components
PPSO	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	Pop size = 50 (for 10, 30, 50 and 100D), w = 0.9 → 0.4, cw = 4 * r * (1 - r), r is a random value and r ∈ (0,1)
TCSPSO	Pop size = 50 (for 10, 30, 50 and 100D), w = 0.9 → 0.4, c1 = c2 = 2
HCLPSO	Pop size = 40 (for 10, 30, 50 and 100D), w = 0.99 → 0.2, c1 = 2.5 → 0.5, c2 = 0.5 → 2.5, c = 3 → 1.5

and 100-dimensional tests are given for CEC2017. The greatest number of rankings of top 3 are given in boldface. The convergence characteristics graphs of SQPPSO-sono and the compared PSO variants are presented in Figs. 7 and 8. Due to page limitations, only the evolutionary processes of f_n over 50 runs of the 100-dimensional test are presented herein.

To better verify the effectiveness of the NRAS and the SQP method when cooperating with SQPPSO-sono individually, the validations of SQPPSO-sono without using the NRAS and the SQP method were performed. Both validations were tested on CEC2013, CEC2014, and CEC2017 benchmark functions for 30 dimensions. Two ablation experiments were conducted to further illustrate the effectiveness of both methods. The experimental results are presented in Tables 5, 6, and 7.

The experimental results for the 10, 30, and 50-dimensional tests on CEC2013 and CEC2014 are presented in Tables SF-1 to SF-6 in the supplementary file, respectively. Additionally, the experimental results for the 10, 30, 50, and 100-dimensional tests on CEC2017 can be found in Tables SF-7 to SF-10 of the supplementary file. These tables summarize the outcomes based on the minimum, median, and mean values, along with their standard deviations, obtained from the last generation of 50 independent runs for each function. Furthermore, the tables include rankings for the minimum, median, and mean values. The smallest minimum, median, and mean values among all recent PSO-based variants are highlighted in boldface, as are the top 1 rankings for the minimum, median, and mean values. To emphasize reliability and enable effective comparisons, the tables retain additional decimal places, showcasing robust variant performances for certain values, rather than using scientific notation.

The Wilcoxon signed-rank test (WST) was performed for SQPPSO-sono versus PSO-sono, MPSO, PPSO, MPSO, TCSPSO, and HCLPSO. Tables SF-1 to SF-10 of the supplementary file also list the outcomes of the WST at the 5% significance level, when contrasting SQPPSO-sono with PSO-sono, MPSO, PPSO, MPSO, TCSPSO, and HCLPSO. In the tables, '≤' and '≥' indicate that SQPPSO-sono conducts worse or better on f_n than PSO-sono, MPSO, PPSO, MPSO, TCSPSO, and HCLPSO. '<' and '>' indicate that SQPPSO-sono conducts significantly worse or better than the compared variants, and '=' implies that there is no difference in the performances of the compared variants. In most cases, SQPPSO-sono surpasses PSO-sono, MPSO, PPSO, MPSO, TCSPSO, and HCLPSO.

Comparison and analysis

In CEC2013 benchmark functions, $f_1 - f_5$, $f_6 - f_{20}$, and $f_{21} - f_{28}$ are unimodal, multimodal, and composition functions, respectively. In CEC2014 benchmark functions, $f_1 - f_3$, $f_4 - f_{16}$, $f_{17} - f_{22}$, and $f_{23} - f_{30}$ are unimodal, multimodal, hybrid, and composition functions, respectively. In CEC2017 benchmark functions, $f_1 - f_3$, $f_4 - f_{10}$, $f_{11} - f_{20}$, and $f_{21} - f_{30}$ are unimodal, multimodal, hybrid, and composition functions, respectively.

In Table 2, SQPPSO-sono shows the largest number of 1st rankings (7 times) for mean values on the 50-dimensional test; the largest number of 1st rankings (10, 9, and 10 times) for minimum values on the 10, 30, and 50-dimensional tests; the largest number of 1st rankings (9 and 9 times) for median values on the 30 and 50-dimensional tests; the overall number of 1st rankings of SQPPSO-sono for the 30-dimensional test is 25 and is equivalent to HCLPSO; the overall number of 1st rankings of SQPPSO-sono for the 50-dimensional test is the best (26 times), whereas the overall number of 1st rankings of HCLPSO for 10-dimensional test is the best (26 times); SQPPSO-sono only ranks 3rd.

In Table 3, SQPPSO-sono has the largest number of 1st rankings for mean, minimum, and median values on the 30 and 50-dimensional tests; thus, the overall number of 1st rankings (both 28 times) of SQPPSO-sono on the 30 and 50-dimensional tests are also the best. HCLPSO has the overall number of 1st rankings (27 times) on the 10-dimensional test, which is the best, while SQPPSO-sono has the second largest number of 1st rankings (25 times).

In Table 4, SQPPSO-sono has the largest number of 1st rankings on all dimensional test except for the rankings (9 times) of mean values, which is the second best, on the 10-dimensional test. The overall number of 1st rankings of SQPPSO-sono for 10, 30, 50, and 100-dimensional tests are 35, 37, 35, and 38 times, respectively, which are the best among all compared algorithms.

Table 2 Statistics of rankings for SQPPSO-sono, PSO-sono, EPSO, PPSO, MPSO, TCSPSO, and HCLPSO on CEC2013 benchmark functions

Dim	Criteria	SQPPSO-sono	PSO-sono	EPSO	PPSO	MPSO	TCSPSO	HCLPSO
10D	Mean [Rankings 1st | 2nd | 3rd]	4| 4| 6	2| 4| 4	7|11| 2	7| 0| 3	1| 1| 2	2| 0| 4	**12**| 6| 6
	Min [Rankings 1st | 2nd | 3rd]	**10**| 5| 3	7| 4| 5	9| 4| 1	7| 3| 2	2| 2| 2	6| 1| 3	5| 3| 8
	Median [Rankings 1st | 2nd | 3rd]	7| 4| 6	3| 6| 3	**9**| 9| 1	7| 3| 4	2| 0| 3	3| 1| 2	**9**| 6| 7
	TOTAL [Rankings 1st | 2nd | 3rd]	21|13|15	12|14|12	25|**24**| 4	21| 6| 9	5| 3| 7	11| 2| 9	**26**|15|**21**
30D	Mean [Rankings 1st | 2nd | 3rd]	7| 0| 1	1| 7| 3	4|13| 2	6| 3| 8	0| 1| 1	1| 0| 5	**9**| 4| 9
	Min [Rankings 1st | 2nd | 3rd]	**9**| 4| 6	4| 5| 4	5|11| 2	7| 2| 6	0| 1| 2	1| 1| 3	7| 2| 4
	Median [Rankings 1st | 2nd | 3rd]	**9**| 1| 4	4| 6| 3	3|11| 3	7| 5| 5	0| 1| 0	1| 0| 3	**9**| 2| 8
	TOTAL [Rankings 1st | 2nd | 3rd]	**25**| 5|11	9|18|10	12|**35**| 7	20|10|19	0| 3| 3	3| 1|11	**25**| 8|**21**
50D	Mean [Rankings 1st | 2nd | 3rd]	**7**| 2| 2	2| 3| 2	4|13| 7	7| 3| 6	0| 0| 3	1| 1| 2	**7**| 6| 6
	Min [Rankings 1st | 2nd | 3rd]	**10**| 3| 4	3| 5| 4	4| 9| 7	7| 4| 5	0| 2| 2	0| 0| 2	7| 5| 2
	Median [Rankings 1st | 2nd | 3rd]	**9**| 1| 3	4| 4| 4	4|11| 7	5| 5| 6	0| 1| 1	1| 0| 4	7| 4| 5
	TOTAL [Rankings 1st | 2nd | 3rd]	**26**| 6| 9	9|12|10	12|**33**|**21**	19|12|17	0| 3| 6	2| 1| 8	21|15|13

Table 3 Statistics of rankings for SQPPSO-sono, PSO-sono, EPSO, PPSO, MPSO, TCSPSO, and HCLPSO on CEC2014 benchmark functions

Dim	Criteria	SQPPSO-sono	PSO-sono	EPSO	PPSO	MPSO	TCSPSO	HCLPSO
10D	Mean [Rankings 1st | 2nd | 3rd]	6| 3| 4	1| 8| 3	4|12| 3	5| 1| 3	0| 1| 6	2| 1| 4	**12**| 5| 6
	Min [Rankings 1st | 2nd | 3rd]	**13**| 4| 5	5| 7| 1	3| 4| 9	7| 0| 1	2| 2| 7	5| 2| 2	4| 8| 2
	Median [Rankings 1st | 2nd | 3rd]	6| 5| 2	1| 8| 3	4|10| 5	7| 0| 4	0| 0| 10	2| 1| 4	**11**| 6| 2
	TOTAL [Rankings 1st | 2nd | 3rd]	25|12|11	7|23| 7	11|**26**|**17**	19| 1| 8	2| 3|23	9| 4|10	**27**|19|10
30D	Mean [Rankings 1st | 2nd | 3rd]	**9**| 3| 2	0| 4| 4	3|12|10	5| 0| 4	0| 2| 2	5| 0| 1	8| 9| 7
	Min [Rankings 1st | 2nd | 3rd]	**9**| 3| 1	3| 4| 8	3|14| 6	7| 0| 2	0| 1| 6	7| 1| 2	3| 5| 8
	Median [Rankings 1st | 2nd | 3rd]	**10**| 4| 2	1| 4| 5	2|11|11	6| 0| 3	0| 2| 1	5| 0| 1	8| 7| 7
	TOTAL [Rankings 1st | 2nd | 3rd]	**28**|10| 5	4|12|17	8|**37**|**27**	18| 0| 9	0| 5| 9	17| 1| 4	19|21|22
50D	Mean [Rankings 1st | 2nd | 3rd]	**9**| 2| 4	1| 2| 5	6| 12| 7	3| 1| 3	0| 0| 1	6| 0| 2	5| 13| 8
	Min [Rankings 1st | 2nd | 3rd]	**9**| 4| 2	1| 3| 5	4| 9| 9	6| 1| 2	0| 4| 3	8| 0| 0	3| 8| 9
	Median [Rankings 1st | 2nd | 3rd]	**10**| 2| 4	1| 3| 6	4|12| 8	3| 1| 3	0| 0| 1	6| 0| 2	6|12| 6
	TOTAL [Rankings 1st | 2nd | 3rd]	**28**| 8|10	3| 8|16	14|**33**|**24**	12| 3| 8	0| 4| 5	20| 0| 4	14|**33**|23

According to the statistics of different dimensional tests on CEC2013, CEC2014, and CEC2017, SQPPSO-sono demonstrated impressive performance on the high-dimensional test and outperformed better performance in most cases. To better review the rankings, the radar maps are given in Fig. 2.

In Table 5, the 30-dimensional test on CEC2013 of SQPPSO-sono performs better on $f_2 - f_3$, $f_6 - f_7$, $f_9 - f_{13}$, $f_{17} - f_{19}$, and $f_{24} - f_{27}$, 16 functions in total, whereas SQPPSO-sono without using the NRAS or the SQP method only performs better on 4 and 8 functions in total, respectively. The 30-dimensional test on CEC2014 of SQPPSO-sono performs better on $f_1 - f_2$, f_4, $f_{13} - f_{15}$, f_{17}, $f_{19} - f_{20}$, f_{22}, $f_{24} - f_{28}$, and f_{30}, 16 functions in total, whereas SQPPSO-sono without using the NRAS or the SQP method only performs better on 7 and 7 functions, respectively. The 30-dimensional test on CEC2017 of SQPPSO-sono performs better on f_2, f_4, f_6, f_8, $f_{11} - f_{12}$, f_{14}, f_{17}, f_{20}, f_{22}, $f_{26} - f_{27}$, and $f_{29} - f_{30}$, 14 functions in total, whereas SQPPSO-sono without using the NRAS or the SQP method only performs better on 10 and 6 functions in total, respectively. The experimental results not only illustrate the effectiveness of the NRAS and the SQP method but also demonstrate the success cooperation of the NRAS and the SQP method and contains the large test suite with all types of functions including unimodal, multimodal, hybrid, and composition functions.

In Table 6, ablation experiments are conducted to better illustrate the effectiveness of the NRAS method. Three fixed pairs of proportions are evaluated for iwPSO and SLPSO: 30–70%, 50–50%, and 70-30%. When compared with these fixed proportions, SQPPSO-sono outperforms with better mean values for multiple CEC2013, CEC2014, and CEC2017 benchmark functions. Specifically, for the 30-70% proportion, SQPPSO-sono performs better in 19, 20, and 20 functions; for the 50-50% proportion, it excels in 20, 22, and 23 functions; and for the 70-30% proportion,

Table 4 Statistics of rankings for SQPPSO-sono, PSO-sono, EPSO, PPSO, MPSO, TCSPSO, and HCLPSO on CEC2017 benchmark functions

Dim	Criteria	SQPPSO-sono	PSO-sono	EPSO	PPSO	MPSO	TCSPSO	HCLPSO
10*D*	Mean [Rankings 1st | 2nd | 3rd]	9| 1| 6	2| 3| 3	5| 14| 8	4| 0| 1	1| 1| 3	1| 0| 4	**12**|10| 4
	Min [Rankings 1st | 2nd | 3rd]	**15**| 4| 3	7| 5| 7	13| 6| 5	6| 1| 1	6| 2| 2	8| 1| 1	11| 2| 4
	Median [Rankings 1st | 2nd | 3rd]	**11**| 4| 3	3| 5| 1	9| 9| 8	5| 1| 2	3| 1| 4	3| 0| 5	9|11| 3
	TOTAL [Rankings 1st | 2nd | 3rd]	**35**| 9|12	12|13|11	27|**29**|**21**	15| 2| 4	10| 4| 9	12| 1|10	32|23|11
30*D*	Mean [Rankings 1st | 2nd | 3rd]	**11**| 0| 6	1| 4| 5	5|14| 6	4| 2| 3	0| 2| 1	0| 0| 0	9| 8| 9
	Min [Rankings 1st | 2nd | 3rd]	**15**| 3| 4	3| 9| 4	8| 5| 8	7| 3| 1	1| 2| 5	0| 0| 0	4| 3| 5
	Median [Rankings 1st | 2nd | 3rd]	**11**| 1| 7	2| 7| 3	7| 10| 6	7| 0| 2	0| 2| 3	0| 0| 0	5| 9| 8
	TOTAL [Rankings 1st | 2nd | 3rd]	**37**| 4|17	6|20|12	20|**29**|20	18| 5| 6	1| 6| 9	0| 0| 0	18|20|**22**
50*D*	Mean [Rankings 1st | 2nd | 3rd]	**10**| 1| 4	3| 4| 3	4|13| 8	6| 2| 3	0| 2| 2	0| 0| 0	7| 8|10
	Min [Rankings 1st | 2nd | 3rd]	**16**| 2| 4	2| 6| 9	7| 7| 4	1| 5| 5	0| 3| 3	0| 1| 0	5| 5| 5
	Median [Rankings 1st | 2nd | 3rd]	**9**| 5| 4	3| 5| 3	6| 8|11	6| 3| 2	0| 1| 3	0| 0| 0	6| 8| 7
	TOTAL [Rankings 1st | 2nd | 3rd]	**35**| 8|12	8|15|15	17|**28**|**23**	13|10|10	0| 6| 8	0| 1| 0	18|21|22
100*D*	Mean [Rankings 1st | 2nd | 3rd]	**11**| 3| 0	7| 0| 3	4|15| 9	3| 6| 3	0| 0| 1	0| 0| 0	5| 6|14
	Min [Rankings 1st | 2nd | 3rd]	**16**| 2| 4	2| 6| 3	4|13| 6	4| 1| 9	0| 0| 4	0| 0| 0	5| 7| 4
	Median [Rankings 1st | 2nd | 3rd]	**11**| 3| 1	7| 0| 3	4|15| 9	3| 6| 3	0| 1| 1	0| 0| 0	5| 5|13
	TOTAL [Rankings 1st | 2nd | 3rd]	**38**| 8| 5	16| 6| 9	12|**43**|24	10|13|15	0| 1| 6	0| 0| 0	15|18|**31**

it achieves better results in 21, 22, and 26 functions. The NARS method shows a significant improvement in terms of efficiency.

In Table 7, ablation experiments were conducted to better illustrate the effectiveness of the SQP method. Two local search mechanisms, the interior point method [45] and CMA-ES [46], were applied for SQPPSO-sono. When compared with the interior point method, SQPPSO-sono performed better in 17, 19, and 21 functions on CEC2013, CEC2014, and CEC2017 benchmark functions, respectively. When compared with CMA-ES, SQPPSO-sono performed better in 15, 17, and 18 functions on CEC2013, CEC2014, and CEC2017 benchmark functions, respectively. Although the SQP method emerged as the winner in pairwise comparisons, it has also provided us with inspiration for developing hybrid local search mechanisms in the future.

In Figs. 7 and 8, SQPPSO-sono produces extremely sharp convergence rates in the very early generations of the optimization process for f_1, $f_3 - f_4$, $f_{10} - f_{15}$, $f_{17} - f_{18}$, f_{25}, f_{28}, and f_{30}. A characteristic of SQPPSO-sono is that it maintains sharp convergence rates in the very earlier (nearly at the beginning of the) generations and keeps almost flat in the later generations of the evolutionary process. However, this feature also has drawbacks. When optimizing some of the problems, it is difficult to jump out once the process falls into local optima. For example, the optimization of f_2, $f_5 - f_6$, f_8, f_{10}, f_{20}, f_{22}, and f_{27} has significant improvement spaces. Due to the early intervention of the SQP method, the algorithm may prematurely fall into local optima on some benchmark functions. One of the research directions is to identify a method that complements the SQP method and works together to form a synergistic mechanism, allowing for a jump-out approach. The local search capability is strongly enhanced in the current search strategy used by SQPPSO-sono; however, the search behavior, particularly keep population diversity in the earlier generations of the run, needs to be improved further in the future.

From Table SF-1, in the 10-dimensional test on CEC2013 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on $f_1 - f_2$ and f_{10}; and the smallest mean value on f_4, the smallest minimum values on f_3, $f_5 - f_8$, f_{16}, and f_{28}, and the smallest median values on $f_3 - f_5$ and f_{27}. From Table SF-4, in the 10-dimensional test on CEC2014 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on $f_1 - f_2$, $f_{17} - f_{18}$, and f_{30}; and the smallest mean and median values on f_{20}; and the smallest minimum values on $f_4 - f_7$ and $f_{25} - f_{28}$. From Table SF-7, in the 10-dimensional test on CEC2017 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on $f_1 - f_2$, f_4, $f_{12} - f_{15}$, and f_{18}; and the smallest mean value on f_{19}; the smallest minimum values on f_3, f_6, f_9, f_{11}, $f_{20} - f_{21}$, and f_{24}; and the smallest median values on f_3, f_9, and f_{11}. In general, SQPPSO-sono is less competitive on CEC2013, CEC2014, and CEC2017 composition functions. However, SQPPSO-sono continues to demonstrate outstanding performance on hybrid functions.

From Table SF-2, in the 30-dimensional test on CEC2013 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on $f_2 - f_4$, $f_6 - f_7$, f_{10}, and f_{16}; and both smallest minimum and median values on f_{26} and f_{28}. From Table SF-5, in the 30-dimensional test

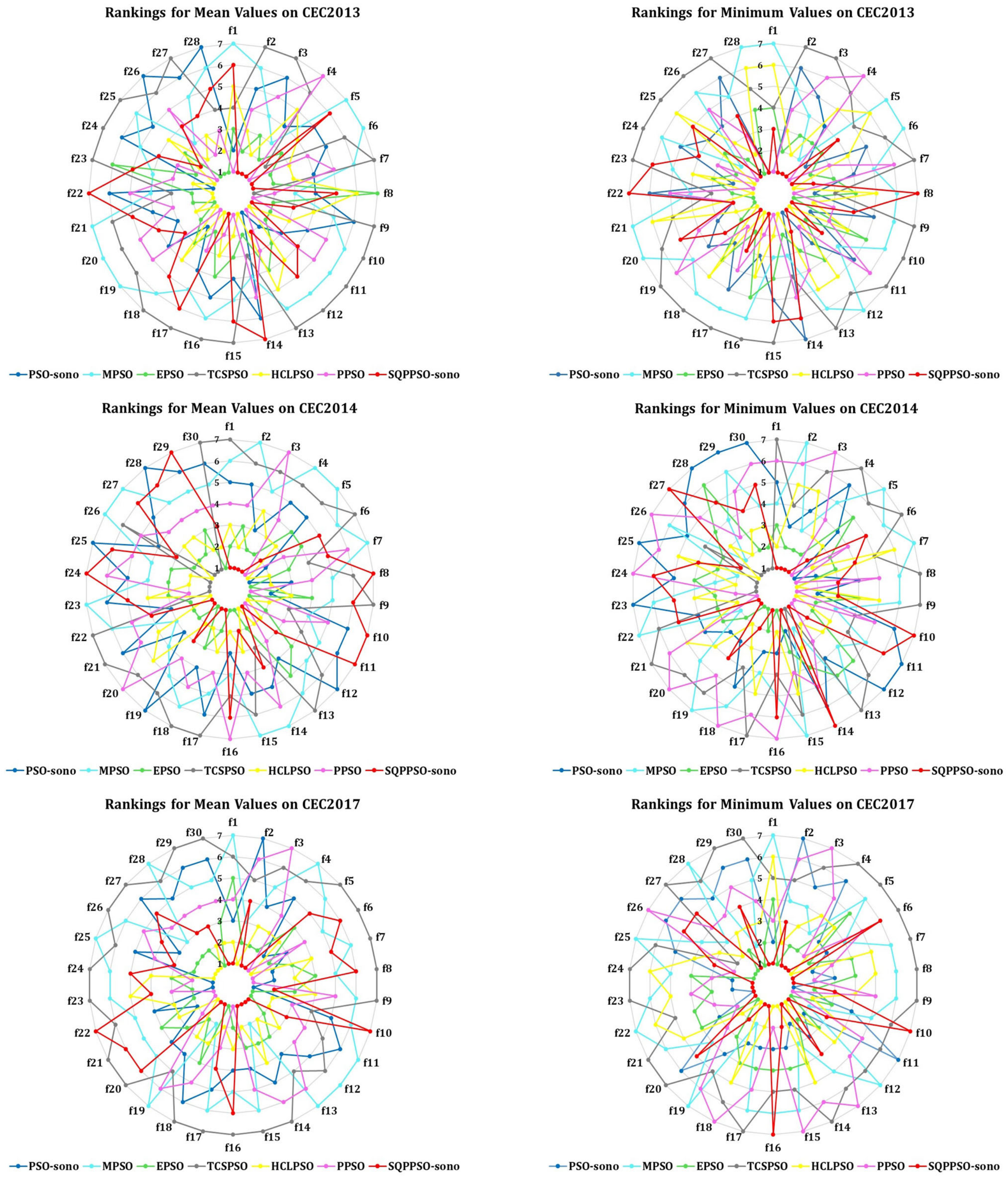

Fig. 2 Radar map of the rankings for mean and minimum values on the 50-dimensional tests. The circumference and radius scale represent the benchmark functions and rankings, respectively

Table 5 The mean values and standard deviations, averaged over 50 runs, of SQPPSO-sono and SQPPSO-sono without using a single component for the 30-dimensional test on CEC2013, CEC2014, and CEC2017 benchmark functions

f_n	SQPPSO-sono			SQPPSO-sono without NRAS			SQPPSO-sono without SQP		
	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017
f_1	3.183231E-13	**5.294078E-04**	4.177884E-03	**0.000000E+00**	5.431410E-04	**1.379589E-03**	5.002221E-14	3.559823E+06	1.182865E+03
	(1.523539E-13)	(2.726823E-04)	(2.008547E-03)	(0.000000E+00)	(1.848963E-04)	(2.191995E-03)	(9.514496E-14)	(4.507360E+06)	(1.374976E+03)
f_2	**1.074544E-03**	**1.474378E-07**	**1.521088E+08**	1.742070E-03	4.891415E-07	1.551712E+16	5.774434E+06	9.430260E-04	1.587328E+13
	(1.279917E-03)	(2.578161E-07)	(7.680573E+08)	(2.860965E-03)	(1.511537E-06)	(8.261538E+16)	(4.433278E+06)	(3.345545E-03)	(1.079220E+14)
f_3	**1.919519E+06**	1.255485E-05	2.673870E-12	8.659336E+06	**1.434615E-06**	**4.729372E-13**	1.342029E+08	1.681903E-01	1.410859E-06
	(3.072112E+06)	(2.245748E-05)	(9.552752E-12)	(1.585156E+07)	(2.336684E-06)	(6.872678E-13)	(3.329746E+08)	(8.693336E-01)	(6.947494E-06)
f_4	5.383161E-03	**2.203518E-11**	**1.952530E+01**	**5.726962E-05**	2.304319E-11	2.842959E+01	2.958381E+01	1.499613E+02	1.183765E+02
	(7.646035E-03)	(2.100077E-11)	(2.825297E+01)	(6.787619E-05)	(3.001718E-11)	(2.991916E+01)	(4.164057E+01)	(6.678908E+01)	(4.584401E+01)
f_5	5.914762E-10	2.000002E+01	7.025171E+01	**1.591616E-14**	**1.999997E+01**	7.164448E+01	1.205080E-13	2.093785E+01	**6.775655E+01**
	(2.217316E-09)	(2.083871E-04)	(3.881191E+01)	(3.984835E-14)	(8.308254E-05)	(4.349370E+01)	(2.727324E-14)	(6.337851E-02)	(1.871807E+01)
f_6	**3.054624E+00**	7.156865E+00	**1.815257E-01**	3.301621E+00	**6.373742E+00**	4.408774E-01	4.768158E+01	1.528799E+01	7.442447E-01
	(8.389096E+00)	(2.163106E+00)	(1.776163E-01)	(8.808561E+00)	(5.537195E+00)	(9.025037E-01)	(3.704468E+01)	(3.448520E+00)	(8.169990E-01)
f_7	**1.410142E+01**	1.098007E-02	9.625433E+01	4.173987E+01	1.214726E-02	**9.468752E+01**	7.190950E+01	**9.598602E-03**	1.700256E+02
	(6.556160E+00)	(2.478490E-02)	(4.159988E+01)	(1.825812E+01)	(1.690332E-02)	(4.176501E+01)	(2.612516E+01)	(9.998966E-03)	(3.504634E+01)
f_8	2.095825E+01	6.238965E+01	**6.744240E+01**	2.095428E+01	6.242359E+01	9.688871E+01	**2.093318E+01**	**5.589668E+01**	7.728821E+01
	(3.933238E-02)	(2.915153E+01)	(3.816537E+01)	(4.647947E-02)	(2.176110E+01)	(3.817503E+01)	(5.018525E-02)	(1.689653E+01)	(2.121254E+01)
f_9	**2.027332E+01**	8.541011E+01	3.518952E+00	2.798469E+01	9.428194E+01	**2.740728E+00**	2.622675E+01	**7.820353E+01**	3.244435E+02
	(3.537344E+00)	(4.015885E+01)	(3.201079E+00)	(2.732475E+00)	(4.478821E+01)	(7.577910E+00)	(3.471354E+00)	(2.214769E+01)	(2.140161E+02)
f_{10}	**2.906205E-02**	3.057761E+03	3.975098E+03	2.930660E-02	2.782020E+03	3.853319E+03	6.630321E-02	**2.707635E+03**	**2.991927E+03**
	(2.079707E-02)	(7.918136E+02)	(6.561889E+02)	(1.873143E-02)	(6.520462E+02)	(6.542441E+02)	(4.628909E-02)	(6.265184E+02)	(5.725531E+02)
f_{11}	**4.385771E+01**	3.653113E+03	**7.345581E+01**	5.342916E+01	3.709961E+03	1.496645E+02	7.527836E+01	**3.284060E+03**	1.411929E+02
	(2.655698E+01)	(6.556731E+02)	(4.637941E+01)	(2.512175E+01)	(6.541564E+02)	(5.468182E+01)	(1.884532E+01)	(6.376462E+02)	(5.043671E+01)
f_{12}	**6.891064E+01**	2.280135E-01	**1.189444E+03**	7.993300E+01	**2.173496E-01**	1.226978E+03	8.560598E+01	1.780424E+00	1.409216E+05
	(4.581231E+01)	(1.740543E-01)	(4.393224E+02)	(4.822190E+01)	(2.361718E-01)	(4.283091E+02)	(2.659955E+01)	(4.084083E-01)	(3.101634E+05)
f_{13}	**9.608956E+01**	**1.948746E-01**	2.738322E+02	1.266050E+02	2.598185E-01	**2.011855E+02**	1.547187E+02	3.730103E-01	4.305559E+04
	(2.661474E+01)	(3.432269E-02)	(1.409271E+02)	(3.648091E+01)	(6.245479E-02)	(1.756429E+02)	(3.180768E+01)	(8.954519E-02)	(2.341587E+05)
f_{14}	3.642618E+03	**3.009491E-01**	**1.027529E+02**	3.098138E+03	3.182522E-01	1.607719E+02	**2.993352E+03**	3.043200E-01	5.569711E+02
	(7.051654E+02)	(5.606041E-02)	(5.888741E+01)	(6.736297E+02)	(5.198598E-02)	(4.732706E+01)	(6.101068E+02)	(8.985813E-02)	(2.118982E+03)

Table 5 continued

f_n	SQPPSO-sono			SQPPSO-sono without NRAS			SQPPSO-sono without SQP		
	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017
f_{15}	4.152745E+03	**5.141379E+00**	1.532183E+02	4.018827E+03	6.464812E+00	**1.522506E+02**	**3.870122E+03**	9.161627E+00	4.790261E+03
	(5.361934E+02)	(2.845243E+00)	(8.634961E+01)	(7.702480E+02)	(3.497855E+00)	(1.157390E+02)	(1.169322E+03)	(4.120619E+00)	(5.729928E+03)
f_{16}	2.449521E-01	1.176210E+01	7.731796E+02	**1.525893E-01**	1.156184E+01	8.389143E+02	2.391447E+00	**1.078816E+01**	**6.896779E+02**
	(3.127478E-01)	(6.500217E-01)	(3.167789E+02)	(8.525538E-02)	(5.940768E-01)	(3.335610E+02)	(2.932708E-01)	(7.977961E-01)	(2.508294E+02)
f_{17}	**1.006279E+02**	**1.303986E+03**	**1.525598E+02**	1.060639E+02	1.325701E+03	2.534838E+02	1.383966E+02	1.663608E+05	2.640935E+02
	(3.842395E+01)	(3.731794E+02)	(9.810202E+01)	(4.442256E+01)	(3.795536E+02)	(1.321675E+02)	(3.165441E+01)	(2.409914E+05)	(1.317730E+02)
f_{18}	**9.922865E+01**	1.037735E+02	1.416505E+02	1.144965E+02	**9.819595E+01**	**1.360831E+02**	1.434805E+02	2.590668E+03	3.902196E+04
	(4.102672E+01)	(3.483826E+01)	(6.138408E+01)	(4.623106E+01)	(3.355800E+01)	(6.186324E+01)	(4.706453E+01)	(3.713814E+03)	(5.821446E+04)
f_{19}	**3.668567E+00**	**8.288240E+00**	1.124892E+02	5.646739E+00	1.082150E+01	**9.785393E+01**	9.350159E+00	9.980589E+00	4.798181E+03
	(1.816943E+00)	(8.672675E+00)	(8.283056E+01)	(3.160853E+00)	(1.233665E+01)	(5.734448E+01)	(4.156150E+00)	(2.617617E+00)	(6.592075E+03)
f_{20}	1.149517E+01	**1.589345E+02**	**2.179955E+02**	1.166995E+01	1.746707E+02	2.958562E+02	**1.136657E+01**	2.009858E+02	2.490727E+02
	(8.065590E-01)	(6.621666E+01)	(1.193498E+02)	(6.039130E-01)	(6.943491E+01)	(1.246262E+02)	(7.729965E-01)	(7.347197E+01)	(1.102759E+02)
f_{21}	3.213215E+02	7.297107E+02	2.768776E+02	3.230632E+02	**6.931327E+02**	2.902668E+02	**3.164506E+02**	2.780423E+04	**2.643726E+02**
	(8.145785E+01)	(2.736371E+02)	(3.463997E+01)	(8.834974E+01)	(3.032746E+02)	(4.249479E+01)	(8.128156E+01)	(8.701869E+04)	(1.796133E+01)
f_{22}	4.396946E+03	**2.110890E+02**	**1.690771E+02**	3.854546E+03	3.403633E+02	2.327403E+02	**3.016865E+03**	2.690029E+02	2.133667E+02
	(8.970435E+02)	(1.125743E+02)	(4.914565E+02)	(7.799812E+02)	(1.501964E+02)	(6.590555E+02)	(6.405453E+02)	(1.450344E+02)	(5.574402E+02)
f_{23}	5.075194E+03	3.152441E+02	4.129704E+02	5.300415E+03	**3.152441E+02**	4.264851E+02	**4.358875E+03**	3.157146E+02	**4.076027E+02**
	(9.579617E+02)	(4.517534E-08)	(3.803898E+01)	(9.102544E+02)	(1.183992E-08)	(4.120508E+01)	(1.019609E+03)	(3.217830E-01)	(2.130262E+01)
f_{24}	**2.496299E+02**	**2.318109E+02**	4.889813E+02	2.638895E+02	2.359475E+02	4.971320E+02	2.711562E+02	2.368611E+02	**4.673674E+02**
	(1.368215E+01)	(7.230683E+00)	(3.733169E+01)	(1.420849E+01)	(8.358762E+00)	(4.244049E+01)	(9.908206E+00)	(8.407270E+00)	(1.490253E+01)
f_{25}	**2.745031E+02**	**2.150196E+02**	3.913742E+02	2.876840E+02	2.222900E+02	**3.909721E+02**	2.829326E+02	2.165566E+02	4.047747E+02
	(8.098488E+00)	(4.507507E+00)	(1.361246E+01)	(9.252209E+00)	(9.631290E+00)	(1.283182E+01)	(8.994982E+00)	(5.014963E+00)	(1.602191E+01)
f_{26}	**2.521200E+02**	**1.002149E+02**	**1.360272E+03**	2.866216E+02	1.002790E+02	1.520042E+03	2.575679E+02	1.103359E+02	1.757309E+03
	(6.741822E+01)	(5.209787E-02)	(5.549606E+02)	(7.848100E+01)	(7.988964E-02)	(7.554939E+02)	(7.758858E+01)	(3.020011E+01)	(7.193779E+02)
f_{27}	**7.363628E+02**	**4.864972E+02**	**5.429710E+02**	8.697969E+02	5.029919E+02	5.492824E+02	9.521817E+02	6.146201E+02	5.470561E+02
	(1.001154E+02)	(8.873804E+01)	(2.005483E+01)	(1.267198E+02)	(1.190008E+02)	(2.002404E+01)	(8.660391E+01)	(1.755556E+02)	(1.908961E+01)
f_{28}	3.638706E+02	**1.111116E+03**	3.477438E+02	4.001806E+02	1.310404E+03	**3.148876E+02**	**3.246095E+02**	1.203625E+03	3.888091E+02
	(2.743361E+02)	(2.512197E+02)	(5.632154E+01)	(3.440304E+02)	(4.505805E+02)	(3.731781E+01)	(2.048382E+02)	(3.366134E+02)	(7.717375E+01)
f_{29}	N/A	4.170298E+06	**6.784628E+02**	N/A	4.869572E+06	8.160405E+02	N/A	**6.776678E+05**	8.031609E+02
	(N/A)	(5.330873E+06)	(1.599943E+02)	(N/A)	(6.066147E+06)	(1.712430E+02)	(N/A)	(3.366153E+06)	(1.757729E+02)
f_{30}	N/A	**2.106136E+03**	**3.200724E+03**	N/A	3.252110E+03	4.233454E+03	N/A	7.479906E+03	1.143282E+04
	(N/A))	(7.922719E+02)	(8.029506E+02)	(N/A)	(1.206452E+03)	(1.214254E+03)	(N/A)	(6.904985E+03)	(7.163931E+03)
Total best	**16**	**16**	**14**	4	7	10	8	7	6

Table 6 Mean values and standard deviations, averaged over 50 runs, of SQPPSO-sono and SQPPSO-sono using different fixed subpopulation ratios for the 30-dimensional test on CEC2013, CEC2014, and CEC2017 benchmark functions

f_n	iwPSO(30%) and SLPSO(70%)			iwPSO(50%) and SLPSO(50%)			iwPSO(70%) and SLPSO(30%)		
	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017
f_1	3.592504E-13	8.961485E-04	8.107760E-03	3.319656E-13	5.484584E-04	5.863113E-03	3.274181E-13	**4.410682E-04**	**1.460628E-03**
	(2.058956E-13)	(2.410762E-04)	(2.712620E-03)	(2.065096E-13)	(1.712804E-04)	(1.349452E-03)	(1.788573E-13)	(1.850895E-04)	(1.083797E-03)
f_2	8.772532E-04	2.220518E-03	2.172032E+07	1.407614E-03	2.285201E-07	3.183696E+10	1.276175E-03	**4.630327E-09**	8.274115E+10
	(9.486985E-04)	(2.359231E-03)	(1.518211E+08)	(2.101697E-03)	(5.654590E-07)	(2.244074E+11)	(2.098268E-03)	(1.533769E-08)	(5.402383E+11)
f_3	7.864611E+06	4.600750E-04	**4.719595E-11**	2.053088E+06	**1.109423E-05**	1.079798E-11	4.501273E+06	**4.560062E-06**	1.022750E-10
	(2.704936E+07)	(3.389759E-04)	(5.011264E-11)	(3.774540E+06)	(1.481751E-05)	(6.473881E-11)	(1.069977E+07)	(6.209155E-06)	(3.646277E-10)
f_4	5.620664E-03	2.567308E+00	2.054550E+01	6.016881E-03	3.238818E+00	**1.517628E+01**	6.747545E-03	2.647040E+00	1.963484E+01
	(6.979205E-03)	(1.270586E+01)	(2.853004E+01)	(6.608761E-03)	(2.290190E+01)	(2.526638E+01)	(6.708118E-03)	(1.270193E+01)	(2.723987E+01)
f_5	1.122810E-09	2.000026E+01	7.092110E+01	**2.537445E-10**	2.000005E+01	7.663152E+01	6.220966E-10	2.000214E+01	9.008086E+01
	(3.492256E-09)	(2.058013E-03)	(3.865386E+01)	(1.010886E-09)	(6.042028E-04)	(3.813147E+01)	(2.212397E-09)	(1.211791E-02)	(3.959383E+01)
f_6	**2.640742E+00**	**4.838739E+00**	**1.254478E-02**	3.092403E+00	7.578360E+00	**1.045378E-01**	4.304919E+00	1.163302E+01	9.374209E-01
	(8.002656E+00)	(1.891078E+00)	(4.770209E-02)	(8.125363E+00)	(1.945302E+00)	(2.259863E-01)	(9.760489E+00)	(2.947107E+00)	(8.342823E-01)
f_7	1.429271E+01	1.773544E-02	1.009562E+02	1.665560E+01	**2.064905E-03**	9.630772E+01	1.628808E+01	**3.985955E-03**	1.060659E+02
	(6.669640E+00)	(2.214528E-02)	(3.949481E+01)	(8.666687E+00)	(8.203203E-03)	(4.148424E+01)	(8.701112E+00)	(8.575239E-03)	(4.117213E+01)
f_8	2.096118E+01	**4.294237E+01**	7.268375E+01	2.096115E+01	**5.651356E+01**	7.870854E+01	**2.093616E+01**	6.686110E+01	8.873014E+01
	(4.596275E-02)	(2.799142E+01)	(3.549420E+01)	(4.782422E-02)	(3.123577E+01)	(3.629612E+01)	(5.271586E-02)	(2.234755E+01)	(3.362936E+01)
f_9	**1.963083E+01**	**7.107706E+01**	**1.702555E+00**	**1.874045E+01**	**7.239300E+01**	4.365799E+00	**1.911457E+01**	8.938681E+01	1.810063E+01
	(4.175727E+00)	(3.805972E+01)	(1.544000E+00)	(3.813828E+00)	(3.773900E+01)	(4.510510E+00)	(3.453929E+00)	(3.445525E+01)	(1.684924E+01)
f_{10}	3.028867E-02	2.964106E+03	4.013482E+03	3.059043E-02	3.122999E+03	4.067660E+03	**2.492238E-02**	**2.660041E+03**	4.115609E+03
	(2.306532E-02)	(7.682562E+02)	(6.409571E+02)	(1.838007E-02)	(6.874363E+02)	(7.713904E+02)	(1.612483E-02)	(6.503486E+02)	(6.761247E+02)
f_{11}	**3.573888E+01**	3.762103E+03	7.461171E+01	**3.438573E+01**	3.677247E+03	1.197703E+02	**3.380865E+01**	3.757131E+03	1.336437E+02
	(2.017423E+01)	(8.522795E+02)	(4.206714E+01)	(2.064474E+01)	(7.338595E+02)	(5.622208E+01)	(2.158556E+01)	(6.514377E+02)	(4.492872E+01)
f_{12}	7.629318E+01	2.375675E-01	1.261544E+03	7.125874E+01	**2.094435E-01**	1.207691E+03	7.434308E+01	**1.917720E-01**	**1.101598E+03**
	(4.658529E+01)	(1.400327E-01)	(4.697132E+02)	(4.154411E+01)	(1.480955E-01)	(4.438850E+02)	(4.302480E+01)	(1.467853E-01)	(4.201323E+02)
f_{13}	1.071926E+02	1.971076E-01	2.824043E+02	1.013869E+02	**1.798504E-01**	2.858924E+02	1.003747E+02	2.138365E-01	3.165452E+02
	(2.869940E+01)	(3.756279E-02)	(2.072923E+02)	(3.167136E+01)	(4.677639E-02)	(1.694338E+02)	(2.943150E+01)	(6.654332E-02)	(1.684099E+02)
f_{14}	**3.550472E+03**	3.258765E-01	1.067576E+02	3.866696E+03	3.146255E-01	1.472013E+02	**3.501442E+03**	3.044936E-01	1.500632E+02
	(6.021598E+02)	(5.454743E-02)	(5.155072E+01)	(5.998550E+02)	(5.142932E-02)	(5.317766E+01)	(7.436656E+02)	(7.370446E-02)	(4.263590E+01)
f_{15}	4.211312E+03	5.837287E+00	1.168754E+02	4.637334E+03	5.576137E+00	**1.388779E+02**	4.321283E+03	5.311752E+00	1.842327E+02
	(6.565416E+02)	(3.087916E+00)	(6.941045E+01)	(7.121486E+02)	(3.160683E+00)	(9.024449E+01)	(7.175676E+02)	(2.877044E+00)	(6.767577E+01)

Table 6 continued

f_n	iwPSO(30%) and SLPSO(70%)			iwPSO(50%) and SLPSO(50%)			iwPSO(70%) and SLPSO(30%)		
	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017
f_{16}	2.498295E-01	1.176888E+01	**6.821775E+02**	**1.960373E-01**	1.179655E+01	**7.440912E+02**	2.539744E-01	**1.129370E+01**	8.503268E+02
	(2.860052E-01)	(6.391487E-01)	(3.685894E+02)	(1.260017E-01)	(4.998684E-01)	(2.965867E+02)	(3.072270E-01)	(7.920472E-01)	(2.808970E+02)
f_{17}	1.023515E+02	1.308567E+03	**8.538954E+01**	**8.611885E+01**	**1.290263E+03**	1.749640E+02	**8.584705E+01**	1.317424E+03	2.503220E+02
	(4.199907E+01)	(3.756155E+02)	(7.544979E+01)	(3.845418E+01)	(4.177580E+02)	(1.168104E+02)	(3.204104E+01)	(3.313638E+02)	(1.390133E+02)
f_{18}	**9.797240E+01**	**8.910216E+01**	1.495523E+02	1.017286E+02	1.040490E+02	1.450286E+02	1.038268E+02	1.173739E+02	1.601404E+02
	(4.410017E+01)	(3.436372E+01)	(6.417722E+01)	(4.686214E+01)	(3.147032E+01)	(5.042814E+01)	(4.113747E+01)	(4.331874E+01)	(6.666056E+01)
f_{19}	4.437812E+00	8.739181E+00	1.187006E+02	3.895615E+00	9.511140E+00	1.150300E+02	**3.622317E+00**	9.030406E+00	1.253964E+02
	(2.802181E+00)	(1.155792E+01)	(8.066429E+01)	(2.041972E+00)	(1.156005E+01)	(6.504947E+01)	(1.630678E+00)	(8.762321E+00)	(4.927833E+01)
f_{20}	1.176016E+01	**1.251659E+02**	**1.634128E+02**	1.164984E+01	1.897147E+02	2.183818E+02	1.175974E+01	1.854098E+02	2.604829E+02
	(6.642791E-01)	(5.704462E+01)	(1.339431E+02)	(7.173261E-01)	(7.473143E+01)	(1.407155E+02)	(5.927352E-01)	(7.132628E+01)	(1.242707E+02)
f_{21}	3.233215E+02	7.462630E+02	**2.665888E+02**	**3.038379E+02**	**7.064187E+02**	2.813964E+02	3.230632E+02	**7.020406E+02**	2.800195E+02
	(7.962532E+01)	(2.511825E+02)	(3.480653E+01)	(8.264848E+01)	(2.703948E+02)	(3.390276E+01)	(9.063024E+01)	(3.018255E+02)	(3.649913E+01)
f_{22}	4.438492E+03	**1.692623E+02**	2.690484E+02	**3.935149E+03**	2.597831E+02	2.850922E+02	4.447402E+03	2.344164E+02	**1.003906E+02**
	(9.331737E+02)	(1.120681E+02)	(8.633606E+02)	(8.907759E+02)	(1.488667E+02)	(9.208880E+02)	(1.121448E+03)	(1.240259E+02)	(9.959441E-01)
f_{23}	5.080281E+03	3.152441E+02	4.160919E+02	5.143834E+03	3.152441E+02	4.194007E+02	5.421247E+03	3.152441E+02	4.307926E+02
	(9.639102E+02)	(6.733072E-08)	(4.133082E+01)	(9.444591E+02)	(3.095124E-08)	(3.695262E+01)	(9.231631E+02)	(5.993839E-08)	(3.717446E+01)
f_{24}	**2.432878E+02**	**2.295063E+02**	4.901861E+02	2.513049E+02	2.326839E+02	4.946264E+02	2.600562E+02	2.369702E+02	5.049174E+02
	(1.180057E+01)	(6.026960E+00)	(3.619361E+01)	(1.116005E+01)	(7.734055E+00)	(3.933601E+01)	(1.122438E+01)	(9.018607E+00)	(3.763914E+01)
f_{25}	**2.692244E+02**	**2.134902E+02**	3.931373E+02	2.748651E+02	2.155084E+02	3.917043E+02	2.806898E+02	2.171709E+02	3.916646E+02
	(6.503645E+00)	(4.487983E+00)	(1.725592E+01)	(7.377319E+00)	(4.601552E+00)	(1.338745E+01)	(9.016058E+00)	(6.562599E+00)	(1.460101E+01)
f_{26}	2.551190E+02	1.021578E+02	**1.054097E+03**	**2.449297E+02**	1.021797E+02	1.403664E+03	2.670138E+02	1.002242E+02	1.537328E+03
	(6.568964E+01)	(1.412129E+01)	(4.084568E+02)	(6.636701E+01)	(1.411791E+01)	(4.813160E+02)	(7.343736E+01)	(4.098078E-02)	(6.553974E+02)
f_{27}	**6.401185E+02**	4.914697E+02	**5.341051E+02**	7.517883E+02	5.192751E+02	**5.418426E+02**	8.187629E+02	5.644891E+02	5.604721E+02
	(1.047184E+02)	(8.994447E+01)	(1.332325E+01)	(8.242861E+01)	(1.018909E+02)	(1.880243E+01)	(8.471828E+01)	(1.513126E+02)	(3.096497E+01)
f_{28}	**3.446827E+02**	**1.049153E+03**	3.498332E+02	3.660373E+02	1.118893E+03	**3.278833E+02**	4.720865E+02	1.336782E+03	**3.395912E+02**
	(2.211227E+02)	(2.314930E+02)	(6.367827E+01)	(3.507368E+02)	(2.785446E+02)	(5.087128E+01)	(4.581286E+02)	(3.674264E+02)	(5.988123E+01)
f_{29}	N/A	**3.553107E+06**	**5.598564E+02**	N/A	4.308846E+06	**6.697069E+02**	N/A	4.512431E+06	7.870629E+02
	(N/A)	(5.083766E+06)	(1.191688E+02)	(N/A)	(5.481547E+06)	(1.522850E+02)	(N/A)	(5.421702E+06)	(1.999314E+02)
f_{30}	N/A	2.587382E+03	3.403394E+03	N/A	2.359757E+03	3.304315E+03	N/A	2.462442E+03	3.510688E+03
	(N/A)	(1.016914E+03)	(6.624873E+02)	(N/A)	(8.942055E+02)	(5.292284E+02)	(N/A)	(1.107459E+03)	(7.826515E+02)
> | <	9|**19**	10|**20**	10|**20**	8|**20**	8|**22**	7|**23**	7|**21**	8|**22**	4|**26**

Table 7 Mean values and standard deviations, averaged over 50 runs, of SQPPSO-sono and SQPPSO-sono using different local search mechanisms for the 30-dimensional test on CEC2013, CEC2014, and CEC2017 benchmark functions

f_n	The SQP Method			The Interior Point Method			CMA-ES		
	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017
f_1	3.183230E-13	5.294080E-04	4.177884E-03	3.274180E-13	**4.371970E-04**	**1.958673E-03**	**1.227820E-13**	4.557690E+05	2.413425E+03
	(1.523540E-13)	(2.726820E-04)	(2.008547E-03)	(1.534580E-13)	(9.746670E-05)	(1.638504E-03)	(1.144730E-13)	(7.416781E+05)	(2.800571E+03)
f_2	1.074544E-03	1.474380E-07	1.521088E+08	**4.326390E-04**	**1.766890E-08**	**8.893535E+05**	6.196317E+05	5.743050E+03	6.947129E+08
	(1.279917E-03)	(2.578160E-07)	(7.680573E+08)	(2.899490E-04)	(5.379750E-08)	(2.943114E+06)	(7.919486E+05)	(4.177754E+03)	(4.520881E+09)
f_3	1.919519E+06	1.255480E-05	2.673870E-12	7.007406E+06	**5.742500E-06**	**1.335820E-12**	4.402686E+07	2.749120E+03	3.282349E+02
	(3.072112E+06)	(2.245750E-05)	(9.552750E-12)	(1.390461E+07)	(6.850340E-06)	(5.137710E-12)	(8.179297E+07)	(3.478958E+03)	(4.310539E+02)
f_4	5.383161E-03	2.203520E-11	1.952530E+01	**3.338477E-03**	2.228030E-11	**1.156843E+01**	3.058068E+03	7.397751E+01	7.666669E+01
	(7.646035E-03)	(2.100080E-11)	(2.825297E+01)	(3.970136E-03)	(1.683860E-11)	(2.204828E+01)	(1.327996E+03)	(5.064966E+01)	(4.483912E+01)
f_5	5.914760E-10	2.000002E+01	7.025171E+01	**3.217340E-12**	**1.999998E+01**	8.719787E+01	**1.796250E-13**	2.093892E+01	**1.508383E+01**
	(2.217320E-09)	(2.083870E-04)	(3.881191E+01)	(1.071880E-11)	(3.478190E-05)	(4.113689E+01)	(6.532830E-14)	(5.365612E-02)	(5.158463E+00)
f_6	3.054624E+00	7.156865E+00	1.815257E-01	**1.823643E+00**	7.909686E+00	1.993519E-01	3.343200E+01	**6.162139E+00**	**4.013982E-02**
	(8.389096E+00)	(2.163106E+00)	(1.776163E-01)	(6.346221E+00)	(1.941791E+00)	(2.943450E-01)	(2.971378E+01)	(1.973107E+00)	(8.646193E-02)
f_7	1.410142E+01	1.098007E-02	9.625433E+01	1.438776E+01	1.130134E-02	1.025169E+02	1.665775E+01	**1.232542E-03**	1.536232E+02
	(6.556160E+00)	(2.478490E-02)	(4.159988E+01)	(7.769755E+00)	(1.684994E-02)	(4.006043E+01)	(9.273273E+00)	(3.159476E-03)	(3.652777E+01)
f_8	2.095825E+01	6.238965E+01	6.744240E+01	2.096063E+01	**5.372768E+01**	8.651963E+01	**2.094095E+01**	**8.278058E+00**	**1.585933E+01**
	(3.933238E-02)	(2.915153E+01)	(3.816537E+01)	(5.827710E-02)	(2.763084E+01)	(3.812813E+01)	(5.862287E-02)	(3.708876E+00)	(5.753657E+00)
f_9	2.027332E+01	8.541011E+01	3.518952E+00	**1.900172E+01**	**7.708919E+01**	4.344982E+00	**1.281621E+01**	**1.570046E+01**	4.377041E+00
	(3.537344E+00)	(4.015885E+01)	(3.201079E+00)	(3.895780E+00)	(3.682686E+01)	(3.243030E+00)	(2.477978E+00)	(6.245258E+00)	(4.790458E+00)
f_{10}	2.906205E-02	3.057761E+03	3.975098E+03	3.010299E-02	3.139827E+03	4.106155E+03	1.679035E-01	**1.253860E+02**	4.037721E+03
	(2.079707E-02)	(7.918136E+02)	(6.561889E+02)	(1.654570E-02)	(6.274994E+02)	(6.880222E+02)	(1.111151E-01)	(2.456693E+02)	(2.496688E+03)
f_{11}	4.385771E+01	3.653113E+03	7.345581E+01	**3.848495E+01**	4.020546E+03	8.308819E+01	**1.138233E+01**	5.642230E+03	8.332137E+01
	(2.655698E+01)	(6.556731E+02)	(4.637941E+01)	(2.272536E+01)	(6.086266E+02)	(4.837036E+01)	(6.757585E+00)	(1.343468E+03)	(4.377165E+01)
f_{12}	6.891064E+01	2.280135E-01	1.189444E+03	7.122670E+01	6.699794E-01	1.337711E+03	**2.844532E+01**	2.424144E+00	2.676203E+04
	(4.581231E+01)	(1.740543E-01)	(4.393224E+02)	(3.741985E+01)	(4.540941E-01)	(4.291868E+02)	(3.209689E+01)	(2.705637E-01)	(3.250373E+04)
f_{13}	9.608956E+01	1.948746E-01	2.738322E+02	1.009575E+02	2.003463E-01	4.944683E+02	**8.421466E+01**	**1.324582E-01**	1.509850E+04
	(2.661474E+01)	(3.432269E-02)	(1.409271E+02)	(2.815198E+01)	(5.540240E-02)	(4.319614E+02)	(5.077011E+01)	(2.394404E-02)	(1.299985E+04)
f_{14}	3.642618E+03	3.009491E-01	1.027529E+02	3.713947E+03	3.115895E-01	1.043877E+02	5.182438E+03	3.843972E-01	5.409027E+03
	(7.051654E+02)	(5.606041E-02)	(5.888741E+01)	(8.198745E+02)	(5.325934E-02)	(5.866544E+01)	(2.071790E+03)	(6.478499E-02)	(8.048356E+03)
f_{15}	4.152745E+03	5.141379E+00	1.532183E+02	4.528467E+03	**4.877362E+00**	1.957388E+02	6.713294E+03	**3.504659E+00**	4.271818E+03
	(5.361934E+02)	(2.845243E+00)	(8.634961E+01)	(5.769931E+02)	(3.125224E+00)	(9.755563E+01)	(2.951610E+02)	(2.341919E+00)	(4.575025E+03)

Table 7 continued

f_n	The SQP Method			The Interior Point Method			CMA-ES		
	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017	CEC2013	CEC2014	CEC2017
f_{16}	2.449521E-01	1.176210E+01	7.731796E+02	8.894102E-01	1.177483E+01	**6.842835E+02**	2.453741E+00	1.201826E+01	**1.514702E+02**
	(3.127478E-01)	(6.500217E-01)	(3.167789E+02)	(5.076275E-01)	(6.520300E-01)	(2.882032E+02)	(3.028703E-01)	(2.957540E-01)	(1.736230E+02)
f_{17}	1.006279E+02	1.303986E+03	1.525598E+02	**8.734705E+01**	1.326398E+03	1.830899E+02	1.799027E+02	5.614162E+04	**7.817863E+01**
	(3.842395E+01)	(3.731794E+02)	(9.810202E+01)	(3.977179E+01)	(3.547759E+02)	(1.160330E+02)	(1.059231E+01)	(6.250272E+04)	(6.148492E+01)
f_{18}	9.922865E+01	1.037735E+02	1.416505E+02	**9.035760E+01**	1.056385E+02	1.423873E+02	1.828941E+02	1.511206E+03	8.815367E+04
	(4.102672E+01)	(3.483826E+01)	(6.138408E+01)	(4.227647E+01)	(2.616061E+01)	(6.633867E+01)	(7.708003E+00)	(1.542560E+03)	(4.570230E+04)
f_{19}	3.668567E+00	8.288240E+00	1.124892E+02	3.973743E+00	1.008130E+01	1.292006E+02	**2.735248E+00**	9.039761E+00	5.790510E+03
	(1.816943E+00)	(8.672675E+00)	(8.283056E+01)	(2.332539E+00)	(1.456837E+01)	(8.414086E+01)	(5.985726E-01)	(8.334964E+00)	(6.760737E+03)
f_{20}	1.149517E+01	1.589345E+02	2.179955E+02	1.171198E+01	**1.380408E+02**	**1.924654E+02**	1.207126E+01	2.329925E+03	**6.564627E+01**
	(8.065590E-01)	(6.621666E+01)	(1.193498E+02)	(8.030875E-01)	(5.510032E+01)	(1.263422E+02)	(5.275506E-01)	(2.492267E+03)	(6.986924E+01)
f_{21}	3.213215E+02	7.297107E+02	2.768776E+02	**3.109671E+02**	**6.484013E+02**	2.802596E+02	**2.883544E+02**	2.482242E+04	**2.191138E+02**
	(8.145785E+01)	(2.736371E+02)	(3.463997E+01)	(6.981397E+01)	(2.741320E+02)	(3.395783E+01)	(7.088991E+01)	(1.697679E+04)	(5.271867E+00)
f_{22}	4.396946E+03	2.110890E+02	1.690771E+02	4.584413E+03	2.135863E+02	2.877533E+02	5.473686E+03	**1.319085E+02**	**1.000991E+02**
	(8.970435E+02)	(1.125743E+02)	(4.914565E+02)	(9.142119E+02)	(1.323396E+02)	(9.382475E+02)	(1.907385E+03)	(9.954436E+01)	(4.903698E-01)
f_{23}	5.075194E+03	3.152441E+02	4.129704E+02	5.390712E+03	3.152441E+02	4.156605E+02	6.564337E+03	**2.953795E+02**	**3.643002E+02**
	(9.579617E+02)	(4.517530E-08)	(3.803898E+01)	(8.272379E+02)	(7.146810E-08)	(3.989723E+01)	(3.210587E+02)	(1.666457E+01)	(8.362553E+00)
f_{24}	2.496299E+02	2.318109E+02	4.889813E+02	2.517456E+02	2.321098E+02	5.015258E+02	**2.463505E+02**	**2.000346E+02**	**4.346239E+02**
	(1.368215E+01)	(7.230683E+00)	(3.733169E+01)	(1.098800E+01)	(7.484125E+00)	(3.888169E+01)	(1.126639E+01)	(1.253356E-02)	(6.960379E+00)
f_{25}	2.745031E+02	2.150196E+02	3.913742E+02	2.745751E+02	**2.131212E+02**	**3.878789E+02**	**2.683965E+02**	**2.000000E+02**	3.932760E+02
	(8.098488E+00)	(4.507507E+00)	(1.361246E+01)	(1.011593E+01)	(3.603268E+00)	(6.515257E+00)	(7.183354E+00)	(4.989310E-13)	(1.328295E+01)
f_{26}	2.521200E+02	1.002149E+02	1.360272E+03	**2.470706E+02**	1.021921E+02	1.485133E+03	2.550229E+02	1.061150E+02	**8.802965E+02**
	(6.741822E+01)	(5.209788E-02)	(5.549606E+02)	(6.639982E+01)	(1.411812E+01)	(4.840833E+02)	(6.543873E+01)	(2.396045E+01)	(3.426516E+02)
f_{27}	7.363628E+02	4.864972E+02	5.429710E+02	7.597135E+02	5.314315E+02	5.459439E+02	**6.806413E+02**	4.984827E+02	5.448810E+02
	(1.001154E+02)	(8.873804E+01)	(2.005483E+01)	(8.803343E+01)	(1.179994E+02)	(2.233199E+01)	(9.615818E+01)	(8.236073E+01)	(1.770685E+01)
f_{28}	3.638706E+02	1.111116E+03	3.477438E+02	**3.607928E+02**	1.170958E+03	**3.247043E+02**	**3.146234E+02**	**7.889349E+02**	3.643501E+02
	(2.743361E+02)	(2.512197E+02)	(5.632154E+01)	(2.800641E+02)	(2.955983E+02)	(4.785335E+01)	(1.659145E+02)	(2.572330E+02)	(6.769590E+01)
f_{29}	N/A	4.170298E+06	6.784628E+02	N/A	**1.243541E+06**	**6.566883E+02**	N/A	**1.056729E+06**	**6.367095E+02**
	(N/A)	(5.330873E+06)	(1.599943E+02)	(N/A)	(3.785067E+06)	(1.714911E+02)	(N/A)	(4.355797E+06)	(1.251645E+02)
f_{30}	N/A	2.106136E+03	3.200724E+03	N/A	2.570980E+03	3.920900E+03	N/A	1.028342E+04	1.873001E+04
	(N/A)	(7.922719E+02)	(8.029506E+02)	(N/A)	(9.340298E+02)	(9.195321E+02)	(N/A)	(8.042035E+03)	(2.638487E+04)
> | <	–	–	–	11|**17**	11|**19**	9|**21**	13|**15**	13|**17**	12|**18**

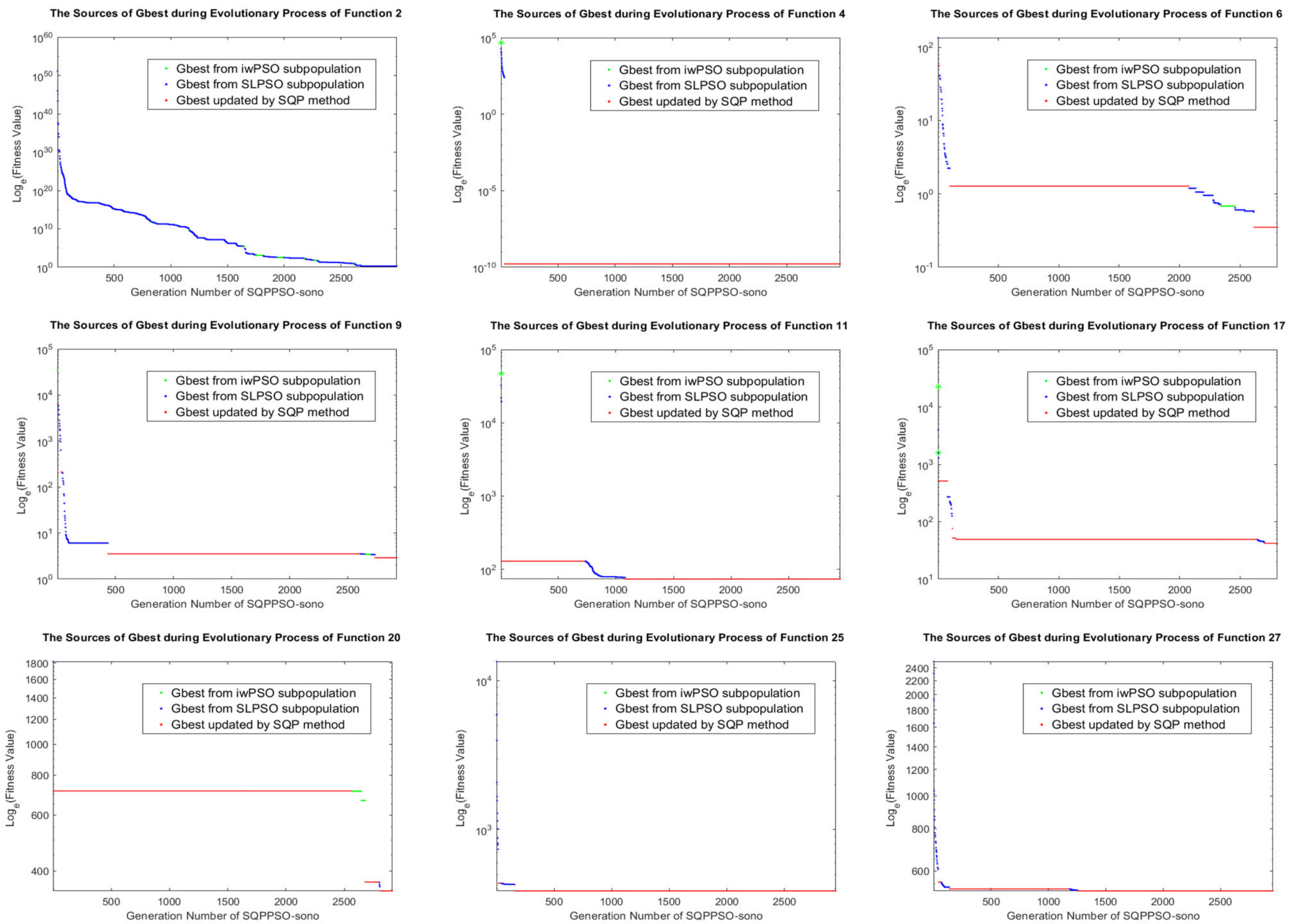

Fig. 3 Sources of best particle in the process of evolution of single run for 30-dimensional test

on CEC2014 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on $f_1 - f_4$, f_{17}, $f_{20} - f_{21}$, and f_{26}; and the smallest minimum value on f_{15}; the smallest mean and median values on f_{18}; and the smallest median value on f_{29}. From Table SF-8, in the 30-dimensional test on CEC2017 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on f_1, $f_3 - f_4$, $f_{12} - f_{15}$, f_{18}, and f_{30}; and the smallest mean values on f_9 and f_{19}; the smallest minimum values on f_2, f_{16}, f_{25}, and f_{26}; and the smallest median and minimum values on f_{22} and f_{28}. Compared with HCLPSO, TCSPSO, and EPSO, SQPPSO-sono is less competitive on CEC2013, CEC2014, and CEC2017 composition functions, respectively. However, the smallest values of SQPPSO-sono cover all types of functions on CEC2013, CEC2014, and CEC2017 benchmark functions.

From Table SF-3, in the 50-dimensional test on CEC2013 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on $f_2 - f_4$, f_6, f_{10}, and f_{16}; and the smallest mean value on f_7; the smallest minimum values on f_{12}, f_{18}, f_{26}, and f_{28}; and the smallest median values on f_7, f_{13}, and f_{28}. From Table SF-6, in the 50-dimensional test on CEC2014 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on $f_1 - f_4$, f_{13}, f_{17}, and $f_{20} - f_{21}$; and the smallest mean value on f_{18}; the smallest minimum value on f_{15}; and the smallest median values on f_{15} and f_{26}. From Table SF-9, in the 50-dimensional test on CEC2017 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on f_1, $f_3 - f_4$, f_{12}, f_{14}, f_{18}, and f_{30}; and the smallest mean values on f_{13}, f_{15}, and f_{19}; the smallest minimum values on f_5, $f_7 - f_8$, f_{17}, $f_{21} - f_{24}$, and f_{28}; and the smallest median values on both f_{13} and f_{19}. In general, the SQPPSO-sono performance on CEC2017 composition functions was remarkably better than that of CEC2013 and CEC2014 composition functions.

From Table SF-10, in the 100-dimensional test on CEC2017 benchmark functions, SQPPSO-sono owns the smallest minimum, median, and mean values on f_1, $f_3 - f_4$, f_{12}, $f_{14} - f_{15}$, f_{18}, and f_{28}; and the smallest mean values on f_{13}, f_{25}, and f_{30}; the smallest minimum values on f_5, $f_7 - f_8$, f_{17}, f_{21}, $f_{23} - f_{24}$, and f_{29}; and the smallest median values on f_{13}, f_{25}, and f_{30}. In general, SQPPSO-sono demonstrates outstanding performance in the 100-dimensional test on all types of functions.

In summary, SQPPSO-sono demonstrated superior performance across unimodal, multimodal, hybrid, and composition functions in the 30, 50, and 100-dimensional tests conducted on CEC2013, CEC2014, and CEC2017 benchmark functions. SQPPSO-sono is less competitive on the 10-dimensional test on CEC2013 and CEC2014. In particular, SQPPSO-sono exhibited better performance on unimodal functions; this can be attributed to the application of the SQP method which amplified the local search capability.

Search behavior of SQPPSO-sono

In Fig. 3, the best fitness values with their sources from a single run are plotted using different colors; green represents the best particle from the iwPSO group; blue represents the best particle from the SLPSO group; red represents the best particle from replacement by a better solution generated by the SQP method. The search modes can be summarized as follows:

- *Both iwPSO and SLPSO conduct an effective search in the whole optimization process.*
- *Either iwPSO, SLPSO, or both conduct effective searches in the very early generations; then, the SQP method conducts an effective search in the later generations and lasts until the end.*
- *Either iwPSO, SLPSO, or both conduct effective searches in the very early generations; then, the SQP method conducts an effective search for a long while, and iwPSO or SLPSO suddenly conducts an effective search for a while. The SQP method later conducts an effective search until the end.*
- *The SQP method conducts an effective search almost at the beginning of the optimization process. Either iwPSO, SLPSO, or both conduct an effective search at some point and cooperate with the SQP method to achieve a sharp convergence rate.*

From the above search modes, the evolutionary process in step descent, like cliff fall, is common; it is consistent with Fig. 7 and 8, and the convergence rate is very sharp. From the figures, in the various evolutionary stages, the SQP method plays an important role in enhancing the local search capability in the very early optimization process and sometimes even at the beginning of the optimization process; thus, it provides an ideal condition to enhance the global search of SQPPSO-sono. In the future, one possible research direction could be to reduce the frequency of usage for the SQP method and increase the usage of the global search method, probably using niching methods [47]. According to the laboratory observations, the interior-point method [48, 49] significantly outperforms the SQP method on a few optimization problems; thus, another possible research direction is to design and ensemble a local search strategy using both the SQP and interior-point methods.

The 2-dimensional tests are conducted for SQPPSO-sono without single methods. The 2-dimensional particles in diverse generations (0, 10, 20, 30) on f_4, f_8, and f_{26} are plotted in Figs. 4, 5, and 6. From the figures, it can be observed that the particles are more scattered during the evolutionary process when SQPPSO-sono does not employ the SQP method; the particles are also more scattered in the later generations for SQPPSO-sono without the SQP method as a local search mechanism. The SQPPSO-sono without the NRAS leads to more gathering of particles due to loss of population diversity, while using the NRAS can maintain the population diversity even in the later generations of the evolution. The figures show forceful evidence that the proposed methods are effective. Thus, the global and local search capabilities of the proposed SQPPSO-sono have been well balanced (Figs. 7, 8).

Algorithmic computational complexity

The computational complexity of the proposed SQPPSO-sono algorithm was determined as delineated in the CEC2017 benchmark competition, which was proposed by [50]. All the experiments were conducted on the below system:

- CPU: Intel Core i7-1165G7 @ 2.80 GHz 1.69 GHz
- RAM: 16GB
- OS: Windows 10
- Software: Matlab 2018b

In Table 8, the computation complexity of the SQPPSO-sono algorithm on 10, 30, 50, and 100 dimensions are presented. In this table, T_0 is the time calculated by performing the below statements:

$x = \text{(double)}0.55;$
for $i = 1 : 1000000$
$x = x + x;\; x = x/2;\; x = x * x;$
$x = \text{sqrt}(x);\; x = \log(x);\; x = \exp(x);\; x = x/(x + 2);$
end

T_1 is the time to perform benchmark function f_{18} individually with 200,000 evaluations in D dimensions. T_2 is the execution time of the SQPPSO-sono algorithm on function f_{18} for 200,000 evaluations with D dimensions. The time ($\hat{T}_2$) is the mean values, averaged over 5 runs, of T_2. As presented in Table 8, T_1, $\hat{T}_2$, and $((\hat{T}_2 - T_1)/T_0)$ scaled linearly with the number of dimensions. The computation complexity comparisons of the SQPPSO-sono algorithm with the compared PSO-based variants are presented in Table 9 for 30 dimensions. The same system and the same procedure are

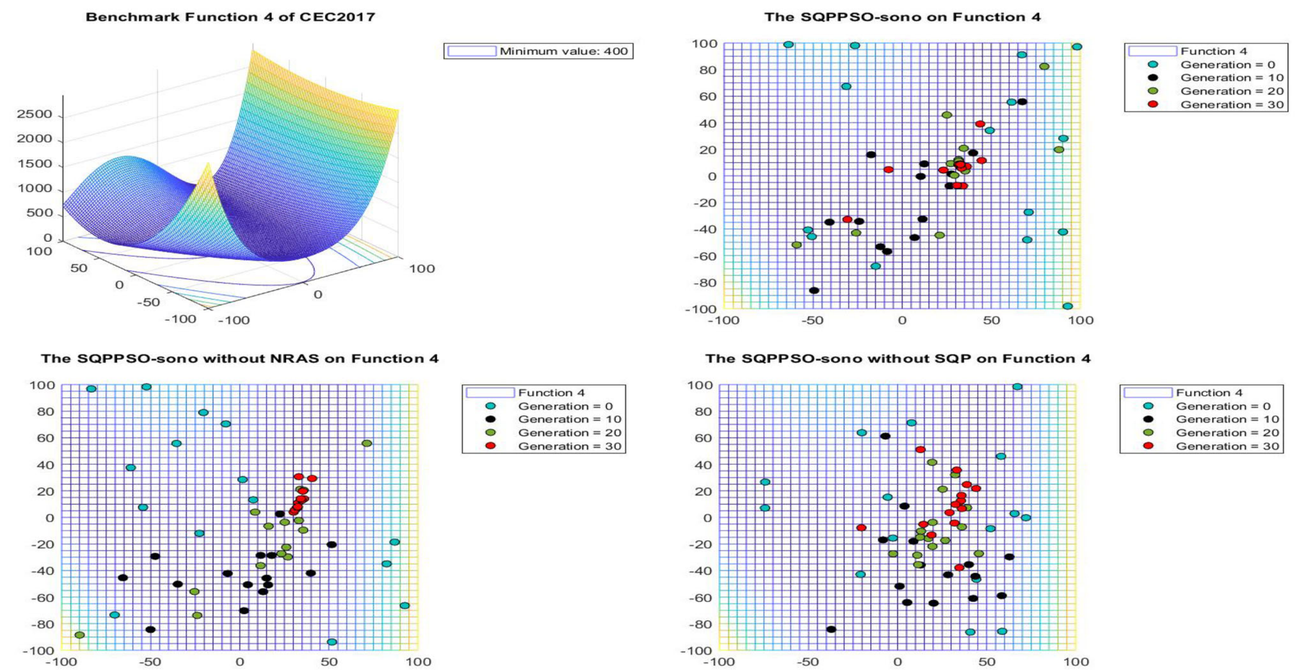

Fig. 4 Plot of 2-dimensional particles in diverse generations (0, 10, 20, 30) on benchmark function 4 of CEC2017

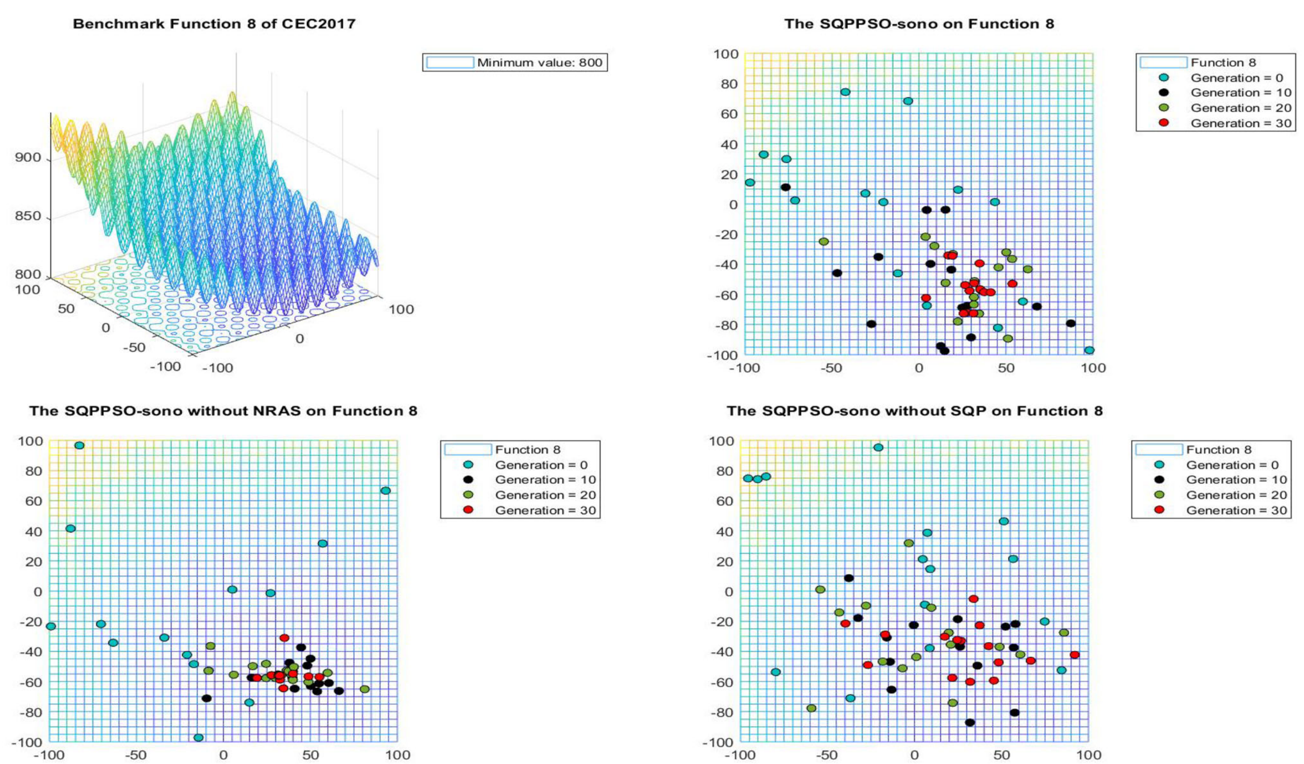

Fig. 5 Plot of 2-dimensional particles in diverse generations (0, 10, 20, 30) on benchmark function 8 of CEC2017

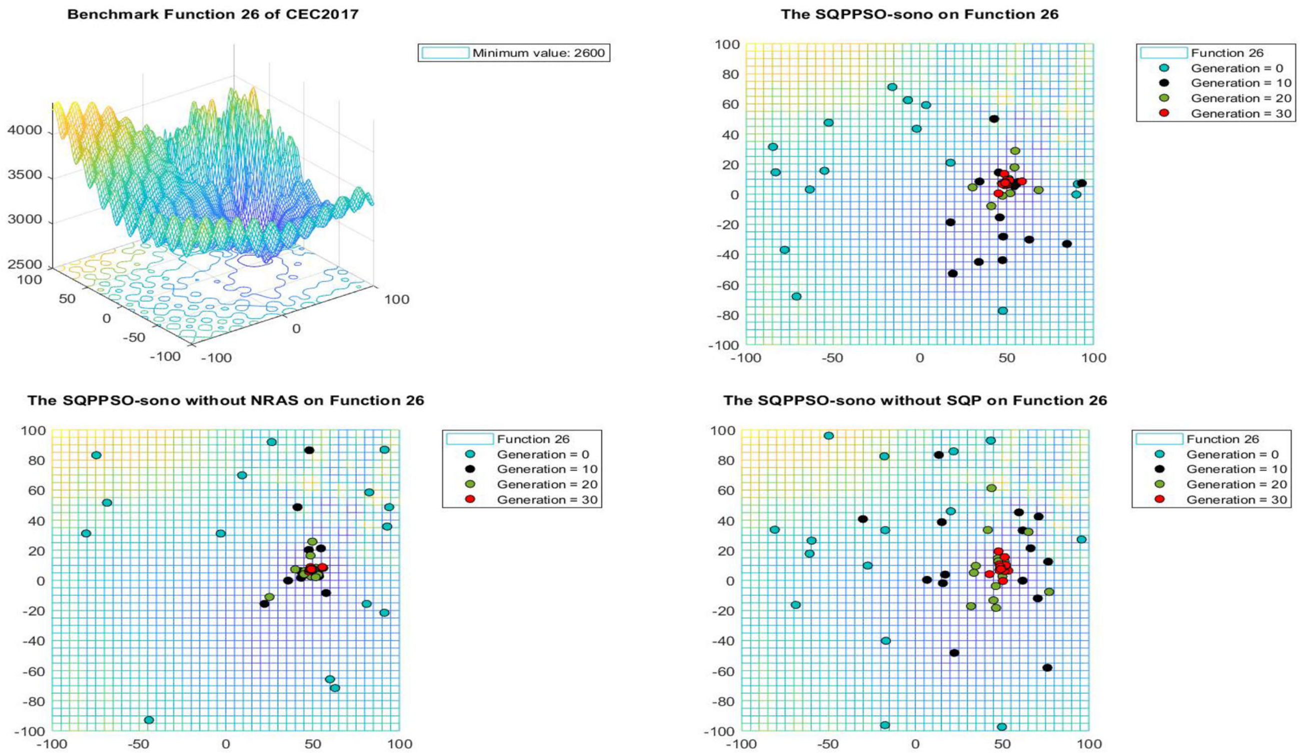

Fig. 6 Plot of 2-dimensional particles in diverse generations (0, 10, 20, 30) on benchmark function 26 of CEC2017

used to calculate, for each compared algorithm, the computation complexity.

In Table 9, the computation complexity of SQPPSO-sono ranks 3rd (excluding SQPPSO-sono without NRAS and SQP), behind PSO-sono and PPSO. The NRAS method is less efficient than the mechanism of the proportional allocation of subpopulations in SQP-sono, as $(\hat{T}_2 - T_1)/T_0$ is only 83.3907 without using the NRAS method. The SQP method is highly efficient than the fully-informed search scheme employed by SQP-sono, as $(\hat{T}_2 - T_1)/T_0$ is 129.5536 and 145.3576 with and without the SQP method. Although the computation cost of SQPPSO-sono is slightly higher than PSO-sono, its performance is much more efficient and reliable than PSO-sono. Such a little performance sacrifice is very worthwhile. Among all variants, the computation cost of SQPPSO-sono is still very competitive.

Conclusions

This paper proposed a sequential quadratic programming (SQP)-based novel strategy for particle swarm optimization on single-objective numerical optimization (SQPPSO-sono). To promote the performance of the strategy, we propose a novel ratio adaptation scheme (NRAS) that divides the subpopulation size considering fitness value and diversity of particles. In SQPPSO-sono, the NRAS method is responsible for dividing subpopulations and the SQP method, replacing fully-informed search scheme, is intermittently invoked to enhance the exploitation capability. The NRAS considers both the quality and diversity of particles during evolution to balance the exploration and exploitation capabilities effectively. Meanwhile, the SQP method is applied to exploit the area around the current global best particle after its generation even in the earlier iterations. The strategic timing and targeted application of this method significantly enhance the algorithm's efficiency. To evaluate the effectiveness of the NRAS and the SQP method, SQPPSO-sono with a single mechanism, either using the NRAS or the SQP method, was validated; the experimental results illustrate that SQPPSO-sono has greater performance with both mechanisms than using individual mechanism. All compared algorithms are tested on CEC2013, CEC2014, and CEC2017 benchmark functions with diverse dimensions. The experimental results demonstrate that SQPPSO-sono has outstanding performance in most cases. The future research directions of SQPPSO-sono include reducing frequency of usage for the SQP method, increasing the global search mechanisms, and designing an ensemble strategy that can dynamically switch according to the landscapes in various evolutionary stages using both the SQP and interior-point methods.

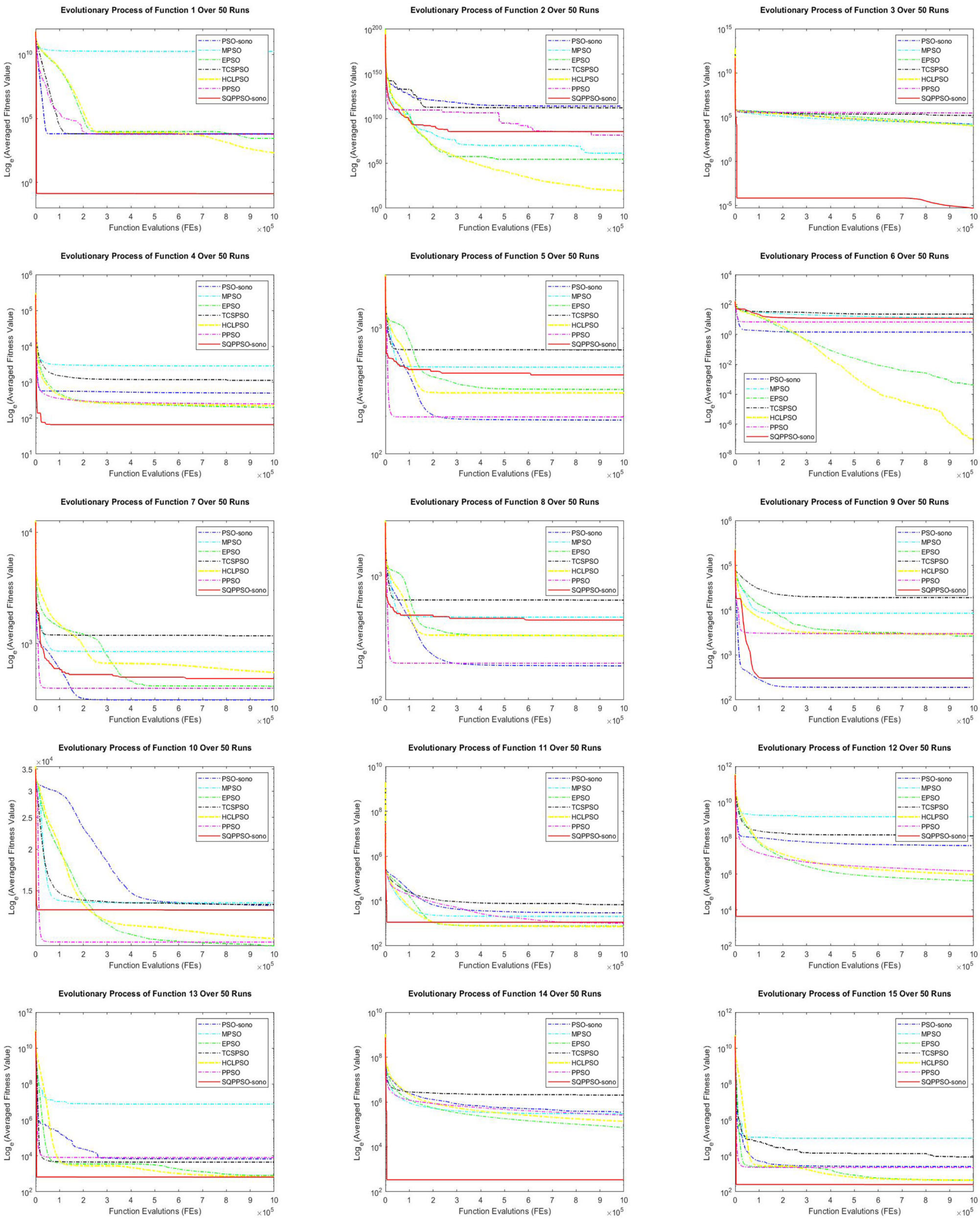

Fig. 7 Evolutionary process of 100D for the proposed SQPPSO-sono algorithm, PSO-sono, EPSO, PPSO, MPSO, TCSPSO, and HCLPSO, averaged over 50 runs for $f_1 - f_{15}$ on CEC2017 benchmark functions. The X-axis represents the function evaluations (FEs) and the Y-axis represents the fitness values of the PSO variants

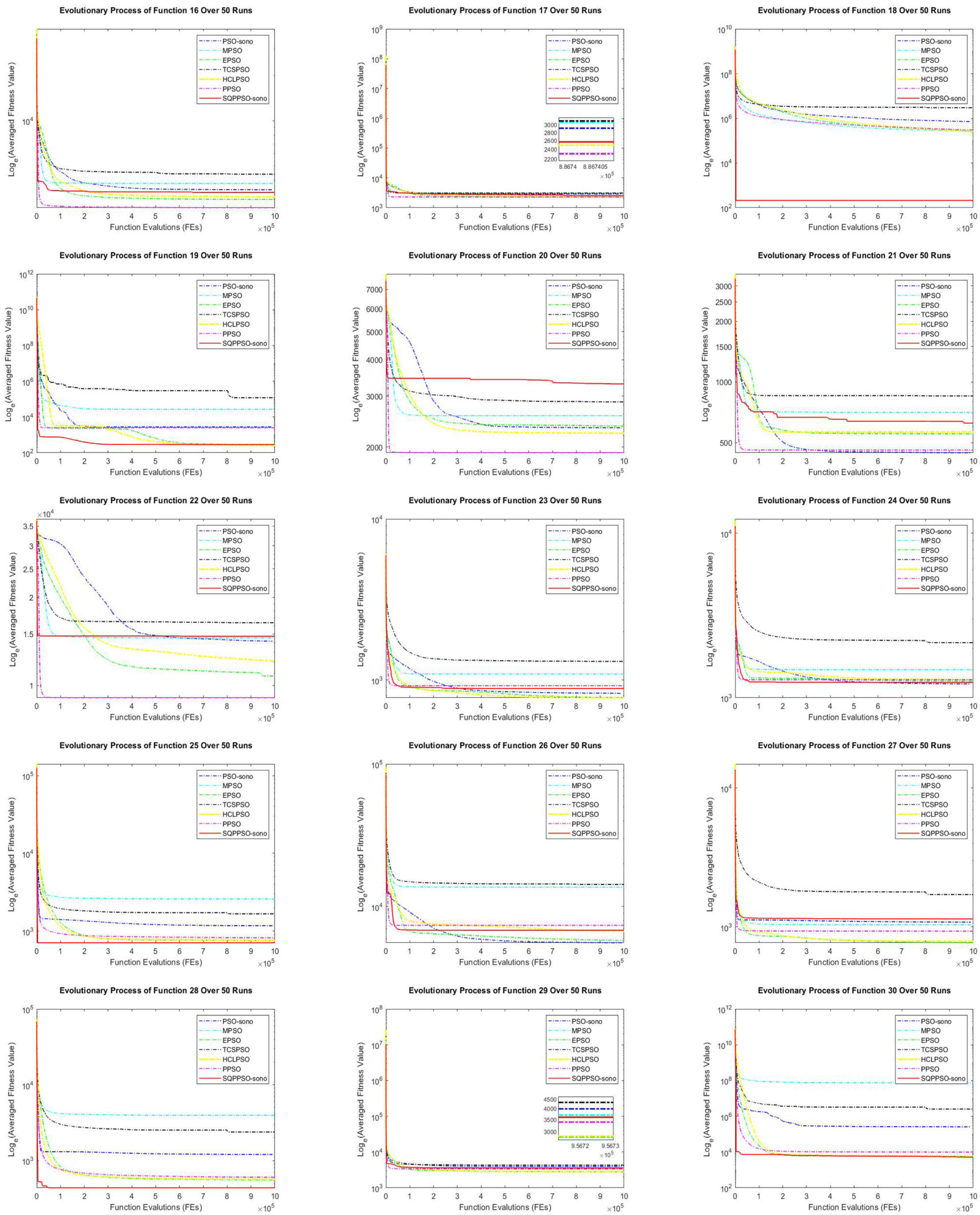

Fig. 8 Evolutionary process of 100D for the proposed SQPPSO-sono algorithm, PSO-sono, EPSO, PPSO, MPSO, TCSPSO, and HCLPSO, averaged over 50 runs for $f_{16} - f_{30}$ on CEC2017 benchmark functions. The X-axis represents the function evaluations (FEs), and the Y-axis represents the fitness values of the PSO variants

Table 8 Computational complexity of the SQPPSO-sono algorithm. The comparison is conducted according to the suggestion of the CEC2017 competition

	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
$10D$	0.0128	0.2970	1.4459	89.5104
$30D$	0.0128	0.7517	2.4147	129.5536
$50D$	0.0128	1.4419	4.4662	235.6047
$100D$	0.0128	5.6440	11.8194	481.0824

Table 9 Computational complexity of SQPPSO-sono, PSO-sono, EPSO, PPSO, MPSO, TCSPSO, and HCLPSO on 30 dimensions. The comparison is conducted according to the suggestion of the CEC2017 competition

30D	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
SQPPSO-sono without NRAS	0.0128	0.7517	1.8191	83.3907
SQPPSO-sono without SQP	0.0128	0.7517	2.6123	145.3576
SQPPSO-sono	0.0128	0.7517	2.4147	129.5536
PSO-sono	0.0128	0.7517	2.0373	100.1527
EPSO	0.0128	0.7517	4.3061	276.8997
PPSO	0.0128	0.7517	1.7580	78.3988
MPSO	0.0128	0.7517	3.7501	233.5902
TCSPSO	0.0128	0.7517	4.2032	268.8832
HCLPSO	0.0128	0.7517	3.5118	215.0224

Acknowledgements We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Funding Funding was provided by The Laboratory Work Research Project of Zhejiang Higher Education Association (Grant number: ZD202201).

References

1. Talbi E-G (2009) Metaheuristics: from design to implementation. Wiley, New York
2. Cura T (2012) A particle swarm optimization approach to clustering. Expert Syst Appl 39(1):1582–1588
3. Sun C, Zeng J, Pan J, Xue S, Jin Y (2013) A new fitness estimation strategy for particle swarm optimization. Inf Sci 221:355–370
4. Boudjelaba K, Ros F, Chikouche D (2014) Potential of particle swarm optimization and genetic algorithms for fir filter design. Circuits Syst Signal Process 33(10):3195–3222
5. Pant S, Anand D, Kishor A, Singh SB (2015) A particle swarm algorithm for optimization of complex system reliability. Int J Perform Eng 11(1):33
6. Liu Q, Wei W, Yuan H, Zhan Z-H, Li Y (2016) Topology selection for particle swarm optimization. Inf Sci 363:154–173
7. Mustafa Servet Kiran (2017) Particle swarm optimization with a new update mechanism. Appl Soft Comput 60:670–678
8. Hajihassani M, Jahed-Armaghani D, Kalatehjari R (2018) Applications of particle swarm optimization in geotechnical engineering: a comprehensive review. Geotech Geol Eng 36(2):705–722
9. Song A, Chen W-N, Tianlong G, Yuan H, Kwong S, Zhang J (2019) Distributed virtual network embedding system with historical archives and set-based particle swarm optimization. IEEE Trans Syst Man Cybern Syst 51(2):927–942
10. Zhang Y, Liu X, Bao F, Chi J, Zhang C, Liu P (2020) Particle swarm optimization with adaptive learning strategy. Knowl Based Syst 196:105789
11. Chen Y, Yan J, Feng J, Sareh P (2021) Particle swarm optimization-based metaheuristic design generation of non-trivial flat-foldable origami tessellations with degree-4 vertices. J Mech Des 143(1)
12. Yang X, Li H, Yu X (2022) Adaptive heterogeneous comprehensive learning particle swarm optimization with history information and dimensional mutation. Multim Tools Appl 1–33
13. Liu F, Zhang J, Liu T (2020) A pso-algorithm-based consensus model with the application to large-scale group decision-making. Complex Intell Syst 6(2):287–298
14. Yilin Yu, Yanli X, Wang F, Li W, Mai X, Hao W (2021) Adsorption control of a pipeline robot based on improved pso algorithm. Complex Intell Syst 7(4):1797–1803
15. Zhi Y, Wang H, Wang L (2022) A state of health estimation method for electric vehicle li-ion batteries using ga-pso-svr. Complex Intell Syst 8(3):2167–2182
16. Abd-El-Atty B (2023) Efficient s-box construction based on quantum-inspired quantum walks with pso algorithm and its application to image cryptosystem. Complex Intell Syst
17. Jie J, Liu K, Zheng H, Wang B, Dai R (2021) High dimensional feature data reduction of multichannel semg for gesture recognition based on double phases pso. Complex Intell Syst 7(4):1877–1893
18. Rao C, Liu Y, Goh M (2023) Credit risk assessment mechanism of personal auto loan based on pso-xgboost model. Complex Intell Syst 9(2):1391–1414

19. Li W, Meng X, Huang Y, Zhang-Hua F (2020) Multipopulation cooperative particle swarm optimization with a mixed mutation strategy. Inf Sci 529:179–196
20. Kao Y-T, Zahara E (2008) A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl Soft Comput 8(2):849–857
21. Kaveh A, Bakhshpoori T, Afshari E (2014) An efficient hybrid particle swarm and swallow swarm optimization algorithm. Comput Struct 143:40–59
22. Moradi P, Gholampour M (2016) A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. Appl Soft Comput 43:117–130
23. Wang F, Zhang H, Li K, Lin Z, Yang J, Shen X-L (2018) A hybrid particle swarm optimization algorithm using adaptive learning strategy. Inf Sci 436:162–177
24. Islam MA, Gajpal Y, ElMekkawy TY (2021) Hybrid particle swarm optimization algorithm for solving the clustered vehicle routing problem. Appl Soft Comput 110:107655
25. Meng Z, Zhong Y, Mao G, Liang Y (2022) Pso-sono: a novel pso variant for single-objective numerical optimization. Inf Sci 586:176–191
26. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In 1998 IEEE international conference on evolutionary computation proceedings. In: IEEE world congress on computational intelligence (Cat. No. 98TH8360). IEEE, 1998, pp 69–73
27. Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. Inf Sci 291:43–60
28. Gill PE, Wong E (2012) Sequential quadratic programming methods. In: Mixed integer nonlinear programming. Springer, Berlin, pp 147–224
29. Richards ZD (2009) Constrained particle swarm optimisation for sequential quadratic programming. Int J Model Identif Control 8(4):361–367
30. Elaiw AM, Xia X, Shehata AM (2012) Dynamic economic dispatch using hybrid de-sqp for generating units with valve-point effects. Math Probl Eng 2012
31. Zhang Y, Yao F, Iu HH-C, Fernando T, Wong KP (2013) Sequential quadratic programming particle swarm optimization for wind power system operations considering emissions. J Mod Power Syst Clean Energy 1(3):227–236
32. Wenxing X, Geng Z, Zhu Q, Xiangbai G (2013) A piecewise linear chaotic map and sequential quadratic programming based robust hybrid particle swarm optimization. Inf Sci 218:85–102
33. Rajabi F, Rezaie B, Rahmani Z (2016) A novel nonlinear model predictive control design based on a hybrid particle swarm optimization-sequential quadratic programming algorithm: Application to an evaporator system. Trans Inst Meas Control 38(1):23–32
34. Boggs PT, Tolle JW (1995) Sequential quadratic programming. Acta Numerica 4:1–51
35. Nocedal J, Wright SJ (2006) Sequential quadratic programming. Numer Optim 529–562
36. Kumar A, Misra RK, Singh D (2017) Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In: 2017 IEEE congress on evolutionary computation, pp 1835–1842
37. Lynn N, Suganthan P (2017) Ensemble particle swarm optimizer. Appl Soft Comput 55:533–548
38. Li T, Jiayi Shi W, Deng, and Zhenda Hu. (2022) Pyramid particle swarm optimization with novel strategies of competition and cooperation. Appl Soft Comput 121:108731
39. Liu H, Zhang X, Liangping T (2020) A modified particle swarm optimization using adaptive strategy. Expert Syst Appl 152:113353
40. Zhang XW, Liu H, Zhang T, Wang QW, Wang Y, LiangPing T (2019) Terminal crossover and steering-based particle swarm optimization algorithm with disturbance. Appl Soft Comput 85:105841
41. Nandar Lynn and Ponnuthurai Nagaratnam Suganthan (2015) Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm Evolut Comput 24:11–24
42. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol 4. IEEE Press, pp 1942–1948
43. Hamidreza Modares and Mohammad-Bagher Naghibi Sistani (2011) Solving nonlinear optimal control problems using a hybrid ipso-sqp algorithm. Eng Appl Artif Intell 24(3):476–484
44. Costa CBB, da Costa AC, Filho RM (2005) Mathematical modeling and optimal control strategy development for an adipic acid crystallization process. Chem Eng Process Process Intensif 44(7):737–753
45. Nemirovski AS, Todd MJ (2008) Interior-point methods for optimization. Acta Numerica 17:191–234
46. Mohamed AW, Hadi AA, Fattouh AM, Jambi KM (2017) Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE, 2017, pp 145–152
47. Shir OM, Bäck T (2005) Niching in evolution strategies. In: Proceedings of the 7th annual conference on genetic and evolutionary computation, pp 915–916
48. Karmarkar N (1984) A new polynomial-time algorithm for linear programming. Combinatorica 4(5):373–395
49. Forsgren A, Gill PE, Wright MH (2002) Interior methods for nonlinear optimization. SIAM Rev 44(4):525–597
50. Awad NH, Ali MZP, Suganthan N, Liang JJ, Qu BY (2017) 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