Cluster Computing (2025)28:16
https://doi.org/10.1007/s10586-024-04785-w

Improved snow ablation optimization for multilevel threshold image segmentation

Rui Zhong[1] · Chao Zhang[2] · Jun Yu[3]

Received: 11 March 2024 / Revised: 11 June 2024 / Accepted: 2 August 2024

Abstract
Snow ablation optimization (SAO) is a novel metaheuristic algorithm (MA). However, we observed certain issues in the original SAO, such as poor capacity in escaping from local optima and slow convergence. To address these limitations, we introduce two strategies: the asynchronous update strategy (AUS) and the top-*k* survival mechanism. We name our proposal SAO$_k$-AUS. In the original SAO, the segregation of search and update delays the improved information sharing, and AUS integrates update processes following each individual's search behavior, facilitating superior knowledge from elites. Additionally, the original SAO adopts an all-acceptance selection principle, maintaining diversity but cannot guarantee the solution quality. Thus, we introduce the top-*k* survival mechanism to ensure the survival of elites. Comprehensive numerical experiments on CEC2013 and CEC2020 benchmark functions, engineering problems, and image segmentation tasks were conducted to evaluate our proposal against eight state-of-the-art MAs. The experimental results and statistical analyses confirm the efficiency of SAO$_k$-AUS. Moreover, the ablation experiments investigate the contribution of two strategies, and we recommend using both proposed strategies simultaneously. The source code of this research is made available in https://github.com/RuiZhong961230/SAO_k-AUS.

Keywords Snow ablation optimization (SAO) · Asynchronous update strategy (AUS) · Top-*k* survival mechanism · Image segmentation

1 Introduction

The exponential growth and maturation of the artificial intelligence (AI) community have led to a dramatic surge in the prevalence of black-box optimization problems across both academic research [1–3] and real-world applications [4–6]. These problems are characterized by intricate fitness landscapes, reflecting the escalating complexity of contemporary optimization tasks [7, 8]. As a result, innovative approaches are needed to navigate the multifaceted search spaces they present effectively. Concurrently, there has been an unprecedented proliferation in the development of new metaheuristic algorithms (MAs) tailored to confront these diverse optimization challenges head-on. This paper does not delve into the rich history and advancement of the MA community, and interested researchers and scholars are encouraged to refer to [9–11] for a comprehensive exploration of this topic.

A newly proposed MA, named snow ablation optimization (SAO) [12], leveraging the snow ablation mathematical model initially formulated by Martinec et al. [13] to simulate snow sublimation and melting phenomena, contributes to the development of the MA community. Extensive numerical experiments, encompassing 29 bound-constrained CEC2017 and 27 real-world constrained CEC2020 benchmark functions and utilizing eight state-of-the-art optimization techniques, have empirically

✉ Jun Yu
yujun@ie.niigata-u.ac.jp

Rui Zhong
rui.zhong.u5@elms.hokudai.ac.jp

Chao Zhang
zhang@u-fukui.ac.jp

[1] Information Initiative Center, Hokkaido University, Sapporo, Japan

[2] Department of Engineering, University of Fukui, Fukui, Japan

[3] Institute of Science and Technology, Niigata University, Niigata, Japan

Published online: 18 October 2024

demonstrated the efficiency and robustness of SAO. Motivated by the remarkable performance of SAO in various optimization tasks, we believe that the expert-designed search patterns and intelligent mechanisms in SAO are promising to solve various optimization tasks. Therefore, this research adopts an SAO-based optimizer to address academic CEC benchmark functions and multilevel threshold image segmentation tasks.

However, while recognizing the potential of SAO, certain aspects of the original SAO algorithm have room for improvement, specifically: (1) selection and acceptance mechanism, and (2) optimum update strategy. Focusing on these two aspects, this paper proposes an enhanced variant of snow ablation optimization, incorporating a top-k survival mechanism and asynchronous update strategy (AUS). The improved algorithm is denoted as SAO$_k$-AUS. In the original SAO, the strict separation of the search operation and update process can lead to optimization hysteresis. Thus, we propose the incorporation of the update process after individual evaluations in an asynchronous manner. This approach enables the prompt updating of the current best optimum and reflects high-level swarm intelligence. Furthermore, we replace the original all-acceptance selection strategy, which accepts offspring individuals entirely, with an efficient top-k survival mechanism. This mechanism retains individuals whose fitness falls within the top-k range to progress to the next iteration. The combination of these two improvements is expected to accelerate the convergence of SAO significantly, which makes the extension of the proposed SAO$_k$-AUS in solving high-dimensional CEC benchmarks and complex multilevel threshold image segmentation tasks possible.

We conducted extensive numerical optimization experiments to investigate the performance of SAO$_k$-AUS on CEC2013 and CEC2020 benchmark functions, and 12 engineering problems listed in [14]. The ablation experiments on CEC2013 benchmark functions are also performed to evaluate the effectiveness of our proposed two strategies independently. Furthermore, we extend our proposed SAO$_k$-AUS to address eight multilevel threshold image segmentation tasks [15, 16]. We compare the performance of our algorithm against eight advanced MAs including differential evolution with self-adaptive populations (SAP-DE) [17], phasor particle swarm optimization (PPSO) [18], modified artificial ecosystem-based optimization (MAEO) [19], tuna swarm optimization (TSO) [20], sea-horse optimization (SHO) [21], serval optimization algorithm (SOA) [22], coati optimization algorithm (COA) [23], and the original snow ablation optimization (SAO) [12]. Our experimental results and statistical analyses demonstrate the competitive performance of SAO$_k$-AUS compared to the benchmarked algorithms. In addition, ablation experiments on CEC2013 benchmark functions provide insights into the individual contribution of our proposed two strategies.

The remainder of this paper is organized as follows. Section 2 presents the related works of the original SAO and the multilevel threshold image segmentation task. Section 3 introduces our proposed SAO$_k$-AUS in detail. Section 4 is dedicated to describing the numerical experiments conducted on various benchmark functions. Section 5 discusses the performance of SAO$_k$-AUS. Finally, Sect. 6 concludes this paper.

2 Related works

2.1 Snow ablation optimization (SAO)

Snow ablation optimization is one of the most recent additions to the family of population-based EAs. SAO draws inspiration from the sublimation and melting phenomena observed in snow, harnessing these natural processes to facilitate both exploitation and exploration behaviors within optimization tasks [12]. Figure 1[1] provides visual representations of the various states of snow that inspire SAO. Like most EAs, SAO can be conceptually divided into three different phases: swarm initialization, exploration phase, and exploitation phase. In the following sections, we will delve into the design and mechanics of these phases, as well as explore SAO's unique feature: the dual population mechanism.

2.1.1 Swarm initialization

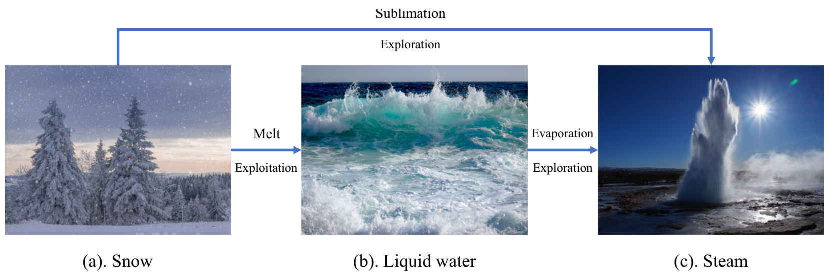
The initial step in SAO is swarm initialization, a critical phase that lays the foundation for the optimization process. As shown in Eq. (1), SAO employs a random placement strategy to distribute the initial swarm across the search space.

$$P = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ x_{31} & x_{32} & \cdots & x_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \tag{1}$$

$$x_{ij} = r_1 \cdot (UB_j - LB_j) + LB_j$$

[1] Pictures are downloaded from https://pixabay.com/ as copyright-free images. (a) https://pixabay.com/photos/winter-landscape-trees-frost-snow-4532412/. (b) https://pixabay.com/photos/wave-splash-ocean-water-sea-5473869/. (c) https://pixabay.com/photos/geyser-strokkur-geyser-strokkur-3242005/.

Fig. 1 The inspiration of SAO is the different states of the snow in nature



where P represents the population, X_i is the ith individual, n and m denote the population size and dimension size, LB_j and UB_j are the lower bound and the upper bound of the search space in jth dimension, and r_1 is a uniform random number ranging from 0 to 1.

2.1.2 Exploration phase

When the snow or liquid water undergoes the transition into steam, the resulting water molecules exhibit irregular movement characterized by high decentralization. This behavior is akin to Brownian motion [24], which serves as a suitable descriptor for such random movement. In SAO, this phenomenon is formalized as an exploration operation, and its mathematical representation is provided in Eq. (2).

$$\begin{aligned} X_i^{t+1} =& Elite + BM \\ &\cdot (r_2 \cdot (X_{best} - X_i^t) + (1 - r_2) \cdot (X_{mean}^t - X_i^t)), \end{aligned} \tag{2}$$

where $Elite$ is selected from the elitism pool containing the best, the second-best, the third-best, and the mean of the top 50% individuals based on uniform distribution. BM is a random vector from 0 to 1 to represent the Brownian motion, r_2 is a random number in the interval (0, 1), X_{best} is the current optimal individual, and X_{mean}^t is the centroid position of all individuals in the tth iteration, calculated using Eq. (3).

$$X_{mean}^t = \frac{1}{n}\sum_{i=1}^{n} X_i^t \tag{3}$$

2.1.3 Exploitation phase

When snow undergoes the melting process and transforms into liquid water, water molecules tend to aggregate. In SAO, this phenomenon is mathematically formulated as an exploitation operator, aimed at promoting the exploration of high-quality elite individuals in the vicinity of the current optimal individual. This operator is represented by Eq. (4).

$$\begin{aligned} X_i^{t+1} =& M \cdot X_{best} + BM \\ &\cdot (r_3 \cdot (X_{best} - X_i^t) + (1 - r_3) \cdot (X_{mean}^t - X_i^t)) \end{aligned} \tag{4}$$

where M is the snowmelt rate, r_3 is a random number in [-1, 1], and the remaining components have the same definitions as in Eq. (2). Additionally, the snowmelt rate M is described using the degree-day method, a well-established mathematical model for snowmelt.

$$M = DDF \cdot T \tag{5}$$

where DDF is the degree-day factor, with a range of [0.35, 0.6] [13], and T represents the average daily temperature. SAO designed a self-adaptive mechanism for DDF based on the iteration times, as formulated in Eq. (6).

$$DDF = 0.35 + 0.25 \cdot \frac{e^{\frac{t}{t_{max}}} - 1}{e - 1} \tag{6}$$

e is a mathematical constant, t means the current iteration time, and t_{max} denotes the maximum iteration time. Additionally, SAO introduces a self-adaptive parameter for T in Eq. (5).

$$T = e^{\frac{-t}{t_{max}}} \tag{7}$$

2.1.4 Dual population mechanism

SAO adopts a dual population mechanism to balance exploitation and exploration. Initially, the population is randomly divided into two equal-sized sub-populations: P_a and P_b. P_a employs an exploration search strategy, as defined in Eq. (2), to generate the offspring, while P_b employs an exploitation search strategy as defined in Eq. (4) for iterative search. As the optimization progresses, the probability of individuals exhibiting irregular, high-decentralized movements increases [12]. Consequently, the size of P_a increases by 1, while the size of P_b decreases by 1 until all individuals are allocated to P_a.

In summary, the pseudocode of SAO is shown in Algorithm 1.

Algorithm 1 SAO [12]

```
Require: Population size: $N$, Dimension: $D$, Maximum iteration: $T$
Ensure: Global optimum: $X_{best}$
1:  Initialize the population $P$ by Eq. (1)
2:  Obtain the current optimum $X_{best}$ from $P$
3:  $t = 0$, $N_a = \mathbf{int}(PS/2)$, $N_b = PS - N_a$
4:  while $t < T$ do
5:      Calculate the snowmelt rate $M$ by Eq. (5)
6:      Randomly separate $P$ to $P_a$ and $P_b$ based on $N_a$ and $N_b$
7:      Obtain $Elite$ solutions with the best, the second-best, the third-best,
    and the mean of the top 50% individuals
8:      for $i = 0$ to $N_a$ do
9:          Generate offspring individual $X_i^{t+1}$ using Eq. (2)
10:     end for
11:     for $i = 0$ to $N_b$ do
12:         Generate offspring individual $X_i^{t+1}$ using Eq. (4)
13:     end for
14:     Evaluate the offspring fitness and update $Elite$ solutions
15:     Obtain the current optimum $X_{best}$ from $P$
16:     $t = t + 1$, $N_a = N_a + 1$, $N_b = N_b - 1$
17: end while
18: return $X_{best}$
```

2.2 Multilevel threshold image segmentation

Image segmentation is a fundamental task in image processing, with applications spanning various domains such as computer vision and remote sensing [25–27]. This task involves partitioning an image into homogeneous regions or segments based on similarities in features, color, texture, and contrast [28, 29]. Multilevel threshold image segmentation is an efficient and straightforward approach used to divide grayscale images into distinct regions, typically consisting of one background and several objects. However, as the number of thresholds increases, the challenge of determining suitable thresholds becomes more pronounced due to the curse of dimensionality [30, 31]. Thus, this remains a significant challenge in the field, necessitating further research.

OTSU algorithm is one of the most classic algorithms to realize image segmentation that operates by minimizing intra-class intensity variance, or equivalently, maximizing the inter-class variance [32]. In the multilevel OTSU algorithm, if we consider the pixel intensity range to be $[0, L-1]$, a set of thresholds t_1, t_2, t_k can be utilized to assign each pixel to one of the multiple object classes $\{C_0, C_1, \ldots, C_k\}$, where

$$\begin{aligned} C_0 &= \{0, 1, \ldots, t_1 - 1\} \\ C_1 &= \{t_1, t_1 + 1, \ldots, t_2 - 1\} \\ &\ldots \\ C_k &= \{t_k, t_k + 1, \ldots, L - 1\} \end{aligned} \tag{8}$$

The probability distributions and means of the classes $\{C_0, C_1, \ldots, C_k\}$ is computed as follows:

$$\begin{aligned} w_0 &= \sum_{i=0}^{t_1-1} p_i \\ w_1 &= \sum_{i=t_1}^{t_2-1} p_i \\ &\ldots \\ w_k &= \sum_{i=t_k}^{L-1} p_i \end{aligned} \tag{9}$$

where $p_i = n_i/N$ is the proportion of pixels with an intensity of i in relation to the total number of pixels, denoted as N. Regardless of the specific threshold values chosen, Eq. (10) is always satisfied:

$$\sum_{i=0}^{k} w_i = w_0 + w_1 + \cdots + w_k = 1 \tag{10}$$

The mean value u_i and the variance σ_i of each component C_i are expressed as follows in Eq. (11).

$$\begin{aligned} u_0 &= \sum_{i=0}^{t_1-1} ip_i/w_0,\ \ \sigma_0 = w_0(u_0 - u_T)^2 \\ u_1 &= \sum_{i=t_1}^{t_2-1} ip_i/w_1,\ \ \sigma_1 = w_1(u_1 - u_T)^2 \\ &\cdots \\ u_k &= \sum_{i=t_k}^{L-1} ip_i/w_k,\ \ \sigma_k = w_k(u_k - u_T)^2 \end{aligned} \tag{11}$$

Here, u_T is the mean value of the entire image, which

$$u_T = \sum_{i=1}^{L-1} ip_i \tag{12}$$

Therefore, the inter-class variance is expressed through Eq. (13)

$$f(t) = \sum_{i=0}^{k} \sigma_i \tag{13}$$

This objective function in the OTSU algorithm serves as the criterion for iteratively searching for a set of suitable thresholds.

2.3 Literature review of SAO

The efficiency and effectiveness of the SAO have garnered significant attention from scholars and researchers since 2023. Numerous enhanced versions of SAO have been developed to address various optimization tasks. For instance, Xiao et al. [33] introduced a multi-strategy boosted snow ablation optimizer (MSAO) to tackle real-world applications. MSAO integrates four efficient search operators: a good point set initialization strategy, a greedy selection method, a differential mutation scheme, and a dynamic lens opposition-based learning strategy. These strategies significantly enhance the performance of the original SAO. Ismaeel et al. [34] focused on the economic load dispatch (ELD) problem, a critical component in power systems, and employed SAO to address this challenge. Their study considered six real-world instances: two cases with 6 generators at loads of 700 MW and 1000 MW, two cases with 10 generators at loads of 1000 MW and 2000 MW, and two cases with 20 generators at loads of 2000 MW and 3000 MW. Comprehensive simulation experiments demonstrated that SAO outperforms all competitor algorithms, highlighting its superiority in real-world scenarios. Jia et al. [35] identified the original SAO's shortcomings, particularly the imbalance between exploration and exploitation and optimization stagnation in high-dimensional instances. They proposed an improved snow ablation optimizer with a heat transfer and condensation strategy (SAOHTC). The heat transfer scheme simulates the transfer of gas molecules from high to low temperatures, moving individuals to locations with better fitness. Additionally, a condensation strategy mimics the transformation of water vapor into water, enhancing the original two-population mechanism. Pandya et al. [36] extended SAO to multi-objective optimization, addressing expansive optimal power flow (OPF) challenges. They developed the multi-objective SAO (MOSAO) by incorporating non-dominated sorting and crowding distance strategies. MOSAO was evaluated on IEEE-30 bus problems, showing significant improvements in convergence, diversity, and distribution attributes. Finally, Lu et al. [37] incorporated mutation schemes from differential evolution (DE) into SAO, proposing a novel DESAO to mitigate the tendency to become trapped in local optima. The performance of DESAO was thoroughly investigated using CEC2017 benchmark functions, demonstrating its effectiveness. As a newly proposed MA, SAO has demonstrated significant potential in solving optimization problems in various domains. This motivates us to propose an improved version of SAO and extend it to solve numerical optimization and multilevel threshold image segmentation tasks.

3 Our proposal: SAO$_k$-AUS

This section provides a detailed introduction to our proposed SAO$_k$-AUS algorithm. We begin by presenting the flowchart of SAO$_k$-AUS in Fig. 2. SAO$_k$-AUS builds upon the main skeleton of SAO while introducing key enhancements. First, SAO$_k$-AUS initializes the population P and various parameters, including the snowmelt rate M, the sizes of sub-populations N_a and N_b, and the elite individuals *Elite*. Like the original SAO, SAO$_k$-AUS randomly divides the population P into two sub-populations, P_a and P_b, based on the size of sub-populations N_a and N_b. Individuals in the sub-population P_a adopt the exploration operation as described in Eq. (2), while individuals in the sub-population P_a focus on exploiting around the current optimum as defined in Eq. (4). One significant difference between SAO$_k$-AUS and SAO lies in the immediate evaluation of offspring individuals. If an offspring solution exhibits a superior fitness value compared to the current optimum, it promptly replaces the current optimum. This rapid update strategy is also employed within the sub-population P_b. Following the evaluation and update process, SAO$_k$-AUS employs a top-k selection mechanism to determine which solutions, among both parent and offspring individuals, should survive to the next generation. SAO$_k$-AUS iterates through the described processes, updating the mentioned parameters, until all available computational resources are exhausted. In the following subsections, we will introduce the proposed AUS and the embedded top-k survival mechanism in detail.

3.1 Asynchronous update strategy

In the original SAO, the process of updating *Elite* is strictly separated from the search operations. This separation is evident in Algorithm 1, where the update process is activated after all offspring individuals are generated (see line 14). However, considering that both the exploration and exploitation operators in Eqs. (2) and (4) utilize the current optimal solution X_{best} to generate offspring individuals, prompt updates for X_{best} can effectively convey the latest global information to the entire swarm. To leverage this advantage, we propose embedding an AUS in SAO$_k$-AUS, and we suggest replacing Algorithm 1 from line 8 to line 13 with Algorithm 2.

This responsive update leverages the superior information discovered so far to construct offspring individuals, which is expected to accelerate optimization convergence and enhance optimization accuracy significantly.

3.2 Top-k survival mechanism

In the original SAO, an all-acceptance principle is employed, similar to the classic particle swarm optimization (PSO). Under this principle, all offspring individuals survive to the next iteration. This approach offers advantages, such as preventing premature optimization and strengthening the algorithm's exploration capacity. However, it tends to decelerate the convergence of optimization, and there's no guarantee that the current optimal solution will survive to the next iteration. To address these limitations, we replace the all-acceptance principle with a more selective top-k survival mechanism. Concretely, in each iteration of optimization, parent individuals and offspring individuals are merged into a single population. From this combined population, only the top-k individuals are selected to survive to the next generation, with k equal to the population size. Algorithm 3 shows the pseudocode of the top-k survival mechanism.

Algorithm 2 Asynchronous update strategy (AUS)

```
Require: Population: $X$, Sub-population size: $N_a$ and $N_b$
Ensure: Offspring: $O$
1: for $i = 0$ to $N_a$ do
2:     Generate offspring individual $X_i^{t+1}$ by Eq. (2)
3:     Evaluate $X_i^{t+1}$ and update $X_{best}$ if $X_i^{t+1}$ has a better fitness value
4: end for
5: for $i = 0$ to $N_b$ do
6:     Generate offspring individual $X_i^{t+1}$ by Eq. (4)
7:     Evaluate $X_i^{t+1}$ and update $X_{best}$ if $X_i^{t+1}$ has a better fitness value
8: end for
9: return $O$
```

Algorithm 3 Top-k survival mechanism

```
Require: Population: $P$, Population fitness: $F_p$, Offspring: $O$, Offspring fitness: $F_o$
Ensure: Survived individuals: $S$
1: $PS \leftarrow \textbf{size}(P)$ # Population size $PS$
2: $O \leftarrow P + O$, $F_o \leftarrow F_p + F_o$ # Merge the population and fitness of the parent and the offspring
3: Simultaneously sort the $F_o$ and $O$ based on $F_o$ in ascend
4: $F_s \leftarrow F_o[0 : PS]$
5: $S \leftarrow O[0 : PS]$
6: return $S$
```

Unlike the all-acceptance criterion in PSO and the one-to-one greedy selection mechanism in DE, the top-k survival mechanism strictly adheres to the "survival of the fittest" principle. This selection mechanism disregards the distinction between parent and offspring individuals; only those with superior fitness can survive. Consequently, this approach is much greedier than one-to-one greedy selection and can further accelerate optimization convergence.

In summary, the pseudocode of the proposed SAO$_k$-AUS is presented in Algorithm 4.

Algorithm 4 SAO$_k$-AUS

```
Require: Population size: $N$, Dimension: $D$, Maximum iteration: $T$
Ensure: Global optimum: $X_{best}$
1: Initialize the population $P$ by Eq. (1)
2: Obtain the current optimum $X_{best}$ from $P$
3: $t = 0$, $N_a = \textbf{int}(PS/2)$, $N_b = PS - N_a$
4: while $t < T$ do
5:     Calculate the snowmelt rate $M$ by Eq. (5)
6:     Randomly separate $P$ to $P_a$ and $P_b$ based on $N_a$ and $N_b$
7:     Obtain $Elite$ solutions with the best, the second-best, the third-best, and the mean of the top 50% individuals
8:     for $i = 0$ to $N_a$ do
9:         Generate offspring individual $X_i^{t+1}$ using Eq. (2)
10:        Adopt asynchronous update strategy to update optimum
11:    end for
12:    for $i = 0$ to $N_b$ do
13:        Generate offspring individual $X_i^{t+1}$ using Eq. (4)
14:        Adopt asynchronous update strategy to update optimum
15:    end for
16:    Evaluate the offspring fitness and update $Elite$ solutions
17:    Adopt top-$k$ survival mechanism to ensure the survival of elites
18:    Obtain the current optimum $X_{best}$ from $P$
19:    $t = t + 1$, $N_a = N_a + 1$, $N_b = N_b - 1$
20: end while
21: return $X_{best}$
```

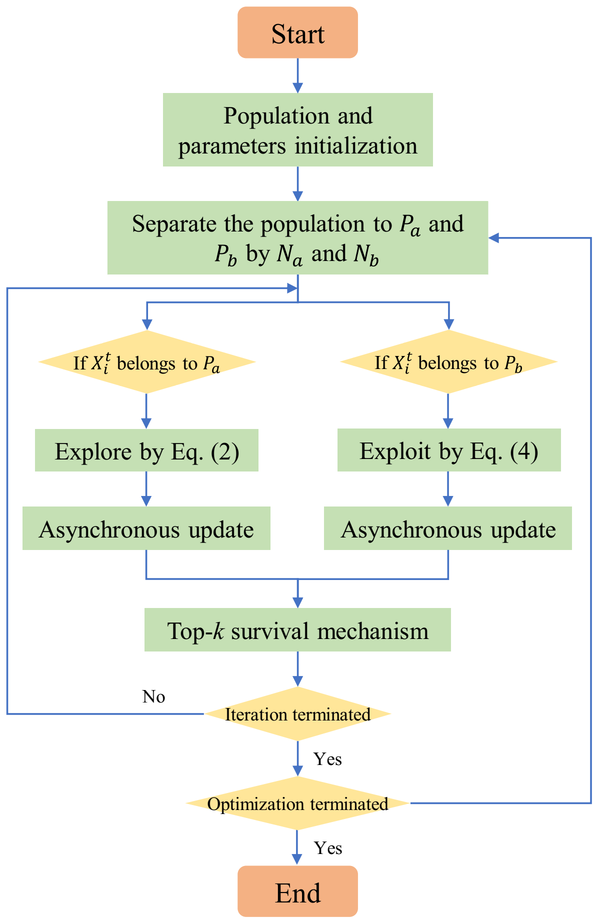

Start
Population and parameters initialization
Separate the population to P_a and P_b by N_a and N_b
If X_i^t belongs to P_a
If X_i^t belongs to P_b
Explore by Eq. (2)
Exploit by Eq. (4)
Asynchronous update
Asynchronous update
Top-k survival mechanism
No
Iteration terminated
Yes
Optimization terminated
Yes
End

Fig. 2 The flowchart of SAO$_k$-AUS

4 Numerical experiments

This section provides a detailed overview of the experimental conditions and presents the corresponding statistical results. Section 4.1 addresses the experiment settings while Sect. 4.2 presents the experimental results and statistical analyses.

4.1 Experiment settings

4.1.1 Benchmark functions

We evaluate the performance of our proposed SAO$_k$-AUS on various benchmark functions, including CEC2013 [38] and CEC2020 [39] benchmark functions. These two benchmark suites are accessed from the OpFuNu library [40]. Additionally, we conducted evaluations on 12 engineering optimization problems presented in [41], which are sourced from the ENOPPY library [42]. In the multilevel threshold segmentation task, eight test images are downloaded from the Berkeley segmentation dataset.[2] The experimental setting involved employing different numbers of thresholds ($nThre$), specifically $\{4, 8, 12\}$.

4.1.2 Competitor algorithms and parameters

In our numerical experiments, eight state-of-the-art MAs serve as competitor algorithms. These methods include SAP-DE [17], PPSO [18], MAEO [19], TSO [20], SHO [21], SOA [22], COA [23], and the original SAO [12]. Except for SAO, the rest competitor algorithms are made available from the MEALPY library [43]. The population size and maximum fitness evaluations for all optimizers in CEC2013 and CEC2020 benchmark functions are fixed at 100 and 1000 $\cdot D$ (D = dimension size), respectively. For engineering optimization problems, the population size and maximum fitness evaluations are set to 100 and 20,000 [44], respectively.

In the multilevel image segmentation task, all algorithms' population size and maximum iteration were set to 30 and 10 $\times nThre$, respectively. To mitigate the impact of randomness in the optimization process, each algorithm was executed with 30 trial runs independently. Furthermore, the internal parameters of these algorithms were configured according to the recommended settings outlined in their respective original papers.

Additionally, 12 engineering optimization problems have constraints, while the competitor algorithms including SAO$_k$-AUS are not inherently designed to handle constrained optimization problems. Thus, it is necessary to incorporate an external penalty function into all optimization techniques. Here, the static penalty function [45] is employed, which is defined by Eq. (14)

$$F(R_i) = f(R_i) + w \cdot \sum_{i=1}^{m} (\max(0, g_i(R_i))) \tag{14}$$

$F(\cdot)$ represents the fitness function, $f(\cdot)$ denotes the objective function, and $g_i(\cdot)$ represents the ith constraint function. The constant w is typically set to $10e^7$ by default.

4.2 Experimental results and statistical analyses

Sections 4.2.1 and 4.2.2 provide the optimization results on CEC2013 and CEC2020 benchmark functions. Section 4.2.3 presents the optimization results on 12 engineering optimization problems, and Sect. 4.2.4 covers the comparative experiments in multilevel threshold image segmentation tasks. The Mann–Whitney U test and the Holm multiple comparison test are employed to determine

[2] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

the statistical significance between every pair of algorithms. Marks $+$, $\approx$, and $-$ represent that our proposed SAO$_k$-AUS is significantly better, has no significant difference, and is significantly worse compared to the other method, while the best fitness value is colored in bold.

4.2.1 Performance investigation on CEC2013

This section shows the experimental results and statistical analyses of optimizers on CEC2013 benchmark functions. Tables 1, 2, 3, and 4 summarize the mean and the standard deviation (std) based on 30 trial runs. Additionally, we conducted ablation experiments to investigate the contribution of each component in the optimization, and these results can be found in Tables 5 and 6.

4.2.2 Performance investigation on CEC2020

This section provides the experimental results and statistical analyses of optimizers on CEC2020 benchmark functions. Tables 7, 8, 9, and 10 summarize the optimization and statistics results on 30 independent trial runs.

4.2.3 Performance on engineering optimization problems

This section presents the experimental results and statistical analyses of optimizers on engineering optimization problems. Tables 11 and 12 summarize the results of all optimization algorithms, while Fig. 3 demonstrates the convergence curves that illustrate the progress of the optimization process.

4.2.4 Performance on image segmentation tasks

Two metrics, peak signal-to-noise ratio (*PSNR*) and structural similarity (*SSIM*), are introduced to evaluate the performance of algorithms. Here are the definitions of these metrics:

$$\begin{aligned} RSME(I,I') &= \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(I_{i,j}-I'_{i,j})^2}{MN}} \\ PSNR(I,I') &= 20\cdot log_{10}\frac{255}{RMSE} \\ SSIM(I,I') &= \frac{(2\mu_I\mu_{I'}+c_1)(2\sigma_{I,I'}+c_2)}{(\mu_I^2+\mu_{I'}^2+c_1)(\sigma_I^2+\sigma_{I'}^2+c_2)} \end{aligned} \tag{15}$$

I and I' represent the original and segmented images, each with dimensions $M \times N$. μ_I denotes the mean intensity of the original image, while $\mu_{I'}$ denotes the mean intensity of the segmented images. Similarly, σ_I and $\sigma_{I'}$ represent the standard deviations of the original and segmented images. $\sigma_{I,I'}$ stands for the covariance between the original and segmented images. Finally, c_1 and c_2 are constants fixed at 0.065.

The root mean square error (*RMSE*) quantifies the average difference between corresponding pixels in the original and segmented images, and a higher *RMSE* signifies greater distortion between the two images. Conversely, a higher *PSNR* indicates a better segmentation process, as it measures the ratio of the maximum possible power of a signal to the power of the noise corrupting the signal. This relationship has been acknowledged in previous studies [46]. *SSIM* assesses the similarity between the original and segmented images across three dimensions: luminance, contrast, and structural content. This metric is designed to better align with human perceptual intuition [15]. Here, we summarize the mean of fitness, *PSNR*, and *SSIM* for eight images in Tables 13, 14, and 15 when the number of thresholds (*nThre*) equals to $\{4, 8, 12\}$. Additionally, we provide segmented images and their corresponding histograms in Figs. 4, 5, 6, and 7.

5 Discussion

This section begins by analyzing the computational complexity. Subsequently, the performance analyses of SAO$_k$-AUS in various optimization tasks are provided.

5.1 Computational complexity analysis of SAO$_k$-AUS

Supposing the population size is N, the dimension size is D, and the maximum iteration is T. SAO$_k$-AUS mainly consists of five steps: swarm initialization, *Elite* solution determination, exploration phase, exploitation phase, and top-k survival mechanism. As the beginning, the computational complexity of the swarm initialization is $O(N \cdot D)$. For elite solution determination, which involves selecting the best, the second-best, the third-best, and the mean of the top 50% individuals, a sort operation is necessary, resulting in a computational complexity is $O(N \log N)$. Similarly, the computational complexity of the top-k survival mechanism is also $O(N \log N)$. The remaining parts are the exploration and the exploitation phase, both with a computational complexity of $O(N \cdot D)$. In summary, the computational complexity of SAO$_k$-AUS is given by $O(N\cdot D+T\cdot(2N\log N+2N\cdot D)) := O(N\cdot D+T\cdot(N\log N+N\cdot D))$.

Table 1 The experimental results and statistical analyses on 30-D CEC2013 benchmark functions

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
f_1	Mean	6.710e+04 +	− 2.713e+02 +	3.628e+03 +	− 9.134e+02 +	1.944e+04 +	4.292e+04 +	2.721e+04 +	8.346e+03 +	**− 1.399e+03**
	Std	8.841e+03	6.020e+02	2.594e+03	1.623e+02	3.761e+03	3.548e+03	6.375e+03	3.755e+03	5.461e+00
f_2	Mean	1.253e+09 +	5.842e+07 +	1.218e+08 +	5.816e+07 +	1.548e+08 +	4.629e+08 +	2.125e+08 +	1.233e+08 +	**7.648e+06**
	Std	4.138e+08	2.564e+07	4.260e+07	1.512e+07	5.280e+07	1.120e+08	7.486e+07	6.047e+07	3.566e+06
f_3	Mean	3.969e+17 +	4.966e+10 +	3.885e+11 +	4.698e+09 +	4.171e+11 +	4.714e+14 +	2.567e+11 +	7.381e+10 +	**2.114e+09**
	Std	1.639e+18	2.909e+10	1.304e+12	2.332e+09	1.034e+12	8.767e+14	2.514e+11	3.011e+10	2.911e+09
f_4	Mean	1.178e+05 +	7.543e+04 +	5.465e+04 +	5.343e+04 +	5.577e+04 +	5.726e+04 +	6.416e+04 +	5.896e+04 +	**8.043e+03**
	Std	2.320e+04	1.615e+04	7.228e+03	5.831e+03	3.927e+03	2.498e+03	6.633e+03	4.624e+03	3.865e+03
f_5	Mean	7.782e+04 +	6.518e+02 +	2.823e+03 +	− 3.342e+02 +	7.636e+03 +	4.432e+04 +	1.394e+04 +	6.559e+03 +	**− 9.995e+02**
	Std	2.443e+04	1.910e+03	1.739e+03	2.264e+02	3.419e+03	1.836e+04	4.851e+03	3.667e+03	1.137e+00
f_6	Mean	1.346e+04 +	− 6.053e+02 +	− 2.709e+02 +	− 7.240e+02 +	8.532e+02 +	6.648e+03 +	1.912e+03 +	− 2.680e+02 +	**− 8.043e+02**
	Std	4.597e+03	1.197e+02	3.670e+02	4.968e+01	4.877e+02	1.327e+03	1.054e+03	2.283e+02	3.392e+01
f_7	Mean	1.826e+08 +	3.583e+06 +	1.156e+06 +	1.551e+05 ≈	4.147e+05 +	1.610e+07 +	2.738e+05 +	2.401e+05 +	**1.408e+05**
	Std	1.852e+08	5.585e+06	2.083e+06	3.466e+04	6.405e+05	1.968e+07	1.012e+05	1.898e+05	3.819e+04
f_8	Mean	− 6.790e+02 +	− 6.790e+02 ≈	− 6.790e+02 ≈	− 6.789e+02 +	− 6.790e+02 ≈	− 6.790e+02 ≈	− 6.790e+02 ≈	− 6.790e+02 ≈	**− 6.790e+02**
	Std	5.202e−02	7.121e−02	3.738e−02	4.281e−02	5.992e−02	6.082e−02	3.438e−02	7.552e−02	5.730e−02
f_9	Mean	− 5.569e+02 +	− 5.611e+02 +	− 5.632e+02 +	− 5.688e+02 +	− 5.645e+02 +	− 5.638e+02 +	− 5.638e+02 +	− 5.688e+02 +	**− 5.716e+02**
	Std	1.302e+00	2.580e+00	2.579e+00	2.885e+00	2.216e+00	1.122e+00	1.514e+00	2.498e+00	2.470e+00
f_{10}	Mean	9.348e+03 +	− 1.017e+01 +	6.170e+02 +	− 2.286e+02 +	2.059e+03 +	5.564e+03 +	2.607e+03 +	8.280e+02 +	**− 4.527e+02**
	Std	1.892e+03	2.762e+02	3.212e+02	9.712e+01	3.738e+02	1.020e+03	7.289e+02	5.006e+02	4.658e+01
f_{11}	Mean	6.690e+02 +	5.852e+01 +	8.711e+01 +	− 1.089e+02 +	1.047e+02 +	3.172e+02 +	1.589e+02 +	− 4.651e+01 +	**− 1.436e+02**
	Std	1.603e+02	9.323e+01	8.898e+01	5.054e+01	5.714e+01	8.781e+01	7.621e+01	6.067e+01	5.725e+01
f_{12}	Mean	7.953e+02 +	1.890e+02 +	2.588e+02 +	1.707e+01 +	1.861e+02 +	3.472e+02 +	2.532e+02 +	5.160e+01 +	**− 6.377e+01**
	Std	1.268e+02	1.094e+02	9.192e+01	5.493e+01	5.092e+01	5.418e+01	6.286e+01	6.123e+01	5.238e+01
f_{13}	Mean	7.938e+02 +	3.617e+02 +	2.833e+02 +	**1.443e+02** ≈	2.845e+02 +	3.903e+02 +	3.180e+02 +	1.812e+02 ≈	1.538e+02
	Std	1.624e+02	9.387e+01	6.289e+01	5.466e+01	4.966e+01	6.179e+01	5.762e+01	6.348e+01	6.121e+01
f_{14}	Mean	8.352e+03 +	5.177e+03 +	6.276e+03 +	5.815e+03 +	6.489e+03 +	6.942e+03 +	7.246e+03 +	5.236e+03 +	**3.573e+03**
	Std	8.352e+03	5.177e+03	6.276e+03	5.815e+03	6.489e+03	6.942e+03	7.246e+03	5.236e+03	3.573e+03

$f_1 - f_5$: unimodal functions; $f_6 - f_{20}$: basic multimodal functions; $f_{21} - f_{28}$: composition functions

Table 2 The experimental results and statistical analyses on 30-D CEC2013 benchmark functions

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
f_{15}	Mean	8.457e+03 +	6.317e+03 +	6.268e+03 +	6.661e+03 +	6.522e+03 +	7.736e+03 +	7.286e+03 +	5.811e+03 +	**4.113e+03**
	Std	3.106e+02	8.144e+02	5.809e+02	5.738e+02	5.743e+02	2.633e+02	4.125e+02	5.706e+02	6.586e+02
f_{16}	Mean	2.033e+02 +	2.023e+02 +	2.020e+02 +	2.029e+02 +	2.018e+02 +	2.030e+02 +	2.021e+02 +	2.015e+02 ≈	**2.012e+02**
	Std	3.146e−01	5.977e−01	6.097e−01	4.219e−01	5.028e−01	3.626e−01	3.806e−01	4.234e−01	6.767e−01
f_{17}	Mean	1.626e+03 +	1.001e+03 +	8.771e+02 +	7.059e+02 +	9.268e+02 +	1.059e+03 +	1.046e+03 +	7.205e+02 +	**6.257e+02**
	Std	2.532e+02	1.010e+02	8.314e+01	4.794e+01	5.375e+01	4.136e+01	8.109e+01	8.249e+01	5.677e+01
f_{18}	Mean	3.307e+03 +	1.698e+03 +	1.386e+03 +	**8.262e+02** −	1.758e+03 +	2.396e+03 +	1.762e+03 +	1.164e+03 +	1.051e+03
	Std	3.498e+02	2.822e+02	1.657e+02	7.281e+01	1.774e+02	1.378e+02	2.610e+02	1.795e+02	1.510e+02
f_{19}	Mean	2.946e+06 +	2.717e+03 +	4.536e+03 +	**5.583e+02** −	2.702e+04 +	3.660e+05 +	1.454e+05 +	7.131e+03 +	1.138e+03
	Std	1.440e+06	5.164e+03	4.106e+03	3.834e+01	1.853e+04	1.613e+05	7.289e+04	8.136e+03	7.807e+02
f_{20}	Mean	6.150e+02 +	6.146e+02 +	6.142e+02 +	6.142e+02 +	6.146e+02 +	6.148e+02 +	6.145e+02 +	6.147e+02 +	**6.133e+02**
	Std	2.261e−07	1.806e−01	4.513e−01	5.311e−01	3.707e−01	1.362e−01	1.464e−01	3.799e−01	9.519e−01
f_{21}	Mean	4.504e+03 +	2.046e+03 +	2.353e+03 +	1.916e+03 ≈	2.576e+03 +	3.147e+03 +	3.167e+03 +	2.528e+03 +	**1.912e+03**
	Std	5.635e+02	1.244e+02	1.307e+02	1.381e+02	1.155e+02	1.393e+02	1.915e+02	1.358e+02	1.031e+02
f_{22}	Mean	9.856e+03 +	6.560e+03 +	7.839e+03 +	7.327e+03 +	8.502e+03 +	8.357e+03 +	8.704e+03 +	7.491e+03 +	**5.419e+03**
	Std	4.611e+02	8.856e+02	7.308e+02	8.175e+02	6.075e+02	3.751e+02	3.343e+02	1.446e+03	9.909e+02
f_{23}	Mean	9.694e+03 +	7.591e+03 +	7.719e+03 +	7.973e+03 +	8.214e+03 +	9.181e+03 +	8.566e+03 +	8.092e+03 +	**5.607e+03**
	Std	3.764e+02	7.654e+02	6.304e+02	4.657e+02	8.156e+02	2.713e+02	4.834e+02	9.831e+02	1.041e+03
f_{24}	Mean	1.356e+03 +	1.319e+03 +	1.313e+03 +	1.298e+03 +	1.309e+03 +	1.365e+03 +	1.311e+03 +	1.292e+03 ≈	**1.290e+03**
	Std	1.770e+01	7.855e+00	7.728e+00	8.334e+00	1.013e+01	1.968e+01	6.827e+00	8.989e+00	1.081e+01
f_{25}	Mean	1.450e+03 +	1.429e+03 +	1.429e+03 +	1.417e+03 ≈	1.446e+03 +	1.485e+03 +	1.427e+03 +	**1.414e+03** ≈	1.415e+03
	Std	9.246e+00	1.337e+01	1.107e+01	1.223e+01	9.694e+00	1.054e+01	7.981e+00	1.358e+01	1.301e+01
f_{26}	Mean	1.607e+03 +	1.593e+03 +	1.588e+03 +	1.553e+03 ≈	1.582e+03 +	1.567e+03 ≈	**1.487e+03** −	1.564e+03 ≈	1.566e+03
	Std	3.506e+01	5.015e+01	4.850e+01	7.348e+01	4.302e+01	8.800e+01	4.721e+01	5.377e+01	4.513e+01
f_{27}	Mean	3.105e+03 +	2.753e+03 +	2.724e+03 +	2.606e+03 ≈	2.726e+03 +	3.358e+03 +	2.710e+03 +	2.565e+03 ≈	**2.558e+03**
	Std	1.333e+02	7.334e+01	9.706e+01	1.297e+02	6.869e+01	1.885e+02	4.781e+01	1.144e+02	1.072e+02
f_{28}	Mean	7.408e+03 +	5.037e+03 +	4.996e+03 +	**3.343e+03** −	4.481e+03 +	5.444e+03 +	5.220e+03 +	4.392e+03 +	4.031e+03
	Std	6.899e+02	5.273e+02	4.224e+02	7.904e+02	2.292e+02	2.852e+02	3.245e+02	2.787e+02	3.120e+02
+/≈/−:		28/0/0	27/1/0	27/1/0	19/6/3	27/1/0	26/2/0	26/1/1	21/7/0	–

While SAO$_k$-AUS theoretically shares identical computational complexity with the original SAO, it may consume more CPU time in practice due to the execution of the sorting operator twice.

5.2 The benefits of two proposed strategies

Our study introduces two innovative strategies aimed at enhancing the efficiency and effectiveness of proposed SAO$_k$-AUS during optimization. The first of these

Table 3 The experimental results and statistical analyses on 50-D CEC2013 benchmark functions

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
f_1	Mean	9.547e+04 +	1.343e+03 +	7.577e+03 +	− 2.637e+01 +	4.243e+04 +	6.749e+04 +	5.976e+04 +	2.452e+04 +	**− 5.317e+02**
	Std	1.107e+04	9.912e+02	2.686e+03	2.905e+02	3.626e+03	4.298e+03	7.613e+03	6.986e+03	1.120e+03
f_2	Mean	3.683e+09 +	1.475e+08 +	2.162e+08 +	1.243e+08 +	3.309e+08 +	1.725e+09 +	7.652e+08 +	2.458e+08 +	**2.592e+07**
	Std	8.693e+08	4.463e+07	7.693e+07	3.149e+07	8.094e+07	4.427e+08	1.653e+08	7.497e+07	9.483e+06
f_3	Mean	5.175e+17 +	6.950e+10 +	1.081e+11 +	1.526e+10 ≈	1.326e+11 +	1.955e+12 +	1.153e+12 +	1.229e+11 +	**1.340e+10**
	Std	1.794e+18	2.880e+10	5.033e+10	6.627e+09	2.293e+10	2.952e+12	2.135e+12	5.596e+10	1.157e+10
f_4	Mean	1.853e+05 +	1.204e+05 +	8.312e+04 +	7.157e+04 +	7.364e+04 +	8.680e+04 +	1.003e+05 +	8.525e+04 +	**2.369e+04**
	Std	5.350e+04	1.951e+04	9.283e+03	8.204e+03	3.894e+03	6.344e+03	9.395e+03	7.726e+03	6.396e+03
f_5	Mean	9.106e+04 +	1.509e+03 +	3.953e+03 +	3.960e+02 +	1.087e+04 +	3.901e+04 +	2.345e+04 +	9.344e+03 +	**− 6.996e+02**
	Std	3.076e+04	1.896e+03	1.603e+03	3.096e+02	1.937e+03	8.692e+03	6.057e+03	2.790e+03	3.357e+02
f_6	Mean	1.283e+04 +	− 4.654e+02 +	− 6.454e+01 +	− 5.493e+02 +	1.882e+03 +	6.490e+03 +	4.156e+03 +	6.180e+02 +	**− 6.602e+02**
	Std	3.803e+03	1.168e+02	1.973e+02	6.595e+01	6.165e+02	1.250e+03	9.504e+02	4.743e+02	6.004e+01
f_7	Mean	3.611e+08 +	4.971e+06 +	1.153e+06 +	4.255e+05 +	6.587e+05 +	1.882e+06 +	1.444e+06 +	5.029e+05 +	**4.034e+05**
	Std	3.586e+08	7.045e+06	1.088e+06	5.760e+04	3.142e+05	1.399e+06	8.168e+05	1.281e+05	1.430e+05
f_8	Mean	− 6.788e+02 +	− 6.788e+02 ≈	− 6.788e+02 ≈	− 6.788e+02 +	− 6.788e+02 ≈	− 6.788e+02 ≈	**− 6.788e+02** ≈	− 6.788e+02 +	− 6.788e+02
	Std	4.839e−02	6.620e−02	3.743e−02	3.285e−02	3.727e−02	4.308e−02	4.986e−02	5.106e−02	4.320e−02
f_9	Mean	− 5.218e+02 +	− 5.281e+02 +	− 5.295e+02 +	− 5.394e+02 ≈	− 5.326e+02 +	− 5.340e+02 +	− 5.311e+02 +	− 5.382e+02 +	**− 5.413e+02**
	Std	1.910e+00	3.586e+00	2.656e+00	4.263e+00	3.652e+00	2.150e+00	2.345e+00	3.935e+00	3.868e+00
f_{10}	Mean	1.674e+04 +	5.267e+02 +	1.341e+03 +	9.312e+01 +	3.786e+03 +	1.055e+04 +	7.089e+03 +	2.543e+03 +	**− 1.551e+02**
	Std	2.667e+03	3.405e+02	3.458e+02	1.050e+02	7.373e+02	1.550e+03	1.303e+03	7.265e+02	1.408e+02
f_{11}	Mean	1.084e+03 +	4.051e+02 +	4.912e+02 +	2.457e+02 +	4.302e+02 +	6.663e+02 +	6.164e+02 +	2.652e+02 +	**8.140e+01**
	Std	2.161e+02	1.116e+02	8.256e+01	7.933e+01	4.429e+01	6.037e+01	7.086e+01	7.848e+01	6.852e+01
f_{12}	Mean	1.311e+03 +	5.697e+02 +	6.223e+02 +	3.782e+02 +	5.553e+02 +	8.168e+02 +	7.687e+02 +	4.198e+02 +	**2.142e+02**
	Std	1.681e+02	1.077e+02	1.131e+02	9.163e+01	6.857e+01	6.522e+01	6.722e+01	6.939e+01	8.485e+01
f_{13}	Mean	1.305e+03 +	7.919e+02 +	6.916e+02 +	5.077e+02 +	6.753e+02 +	8.083e+02 +	7.990e+02 +	5.192e+02 +	**4.494e+02**
	Std	1.609e+02	8.042e+01	7.687e+01	5.974e+01	6.125e+01	4.895e+01	6.606e+01	7.126e+01	7.233e+01
f_{14}	Mean	1.491e+04 +	1.018e+04 +	1.183e+04 +	1.006e+04 +	1.284e+04 +	1.262e+04 +	1.325e+04 +	1.076e+04 +	**6.634e+03**
	Std	5.681e+02	1.236e+03	8.412e+02	1.140e+03	7.882e+02	2.989e+02	4.925e+02	7.836e+02	8.540e+02

Table 4 The experimental results and statistical analyses on 50-D CEC2013 benchmark functions

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO_k-AUS
f_{15}	Mean	1.579e+04 +	1.272e+04 +	1.278e+04 +	1.285e+04 +	1.330e+04 +	1.454e+04 +	1.415e+04 +	1.236e+04 +	**9.054e+03**
	Std	3.843e+02	1.393e+03	8.291e+02	9.972e+02	5.959e+02	4.448e+02	4.372e+02	7.157e+02	1.124e+03
f_{16}	Mean	2.042e+02 +	2.033e+02 +	2.026e+02 +	2.038e+02 +	2.027e+02 +	2.039e+02 +	2.028e+02 +	2.022e+02 +	**2.018e+02**
	Std	5.676e−01	7.993e−01	4.683e−01	3.601e−01	4.298e−01	3.178e−01	4.233e−01	5.274e−01	9.047e−01
f_{17}	Mean	2.337e+03 +	1.601e+03 +	1.474e+03 +	1.165e+03 +	1.403e+03 +	1.604e+03 +	1.623e+03 +	1.188e+03 +	**9.652e+02**
	Std	2.829e+02	1.544e+02	1.027e+02	7.996e+01	7.238e+01	4.929e+01	8.360e+01	9.661e+01	9.952e+01
f_{18}	Mean	4.753e+03 +	3.017e+03 +	2.359e+03 +	**1.319e+03** −	2.899e+03 +	3.564e+03 +	3.118e+03 +	2.122e+03 ≈	1.959e+03
	Std	4.750e+02	2.926e+02	3.065e+02	1.074e+02	1.890e+02	1.845e+02	3.454e+02	2.817e+02	2.640e+02
f_{19}	Mean	2.066e+06 +	4.327e+03 ≈	3.683e+03 ≈	**6.238e+02** −	4.176e+04 +	2.889e+05 +	2.504e+05 +	1.679e+04 +	2.593e+03
	Std	1.619e+06	3.630e+03	2.060e+03	3.121e+01	1.454e+04	9.308e+04	1.179e+05	1.126e+04	1.266e+03
f_{20}	Mean	6.250e+02 +	6.245e+02 +	6.243e+02 +	6.239e+02 +	6.243e+02 +	6.240e+02 +	6.243e+02 +	6.240e+02 +	**6.226e+02**
	Std	1.908e−06	7.818e−02	3.403e−01	6.097e−01	4.441e−01	2.154e−01	3.579e−01	6.398e−01	1.304e+00
f_{21}	Mean	6.365e+03 +	1.784e+03 +	2.222e+03 +	1.443e+03 +	3.590e+03 +	4.389e+03 +	4.667e+03 +	3.417e+03 +	**1.354e+03**
	Std	8.569e+02	3.498e+02	2.657e+02	5.634e+01	2.003e+02	9.965e+01	2.277e+02	2.860e+02	1.655e+02
f_{22}	Mean	1.723e+04 +	1.297e+04 +	1.463e+04 +	1.229e+04 +	1.566e+04 +	1.495e+04 +	1.556e+04 +	1.350e+04 +	**9.620e+03**
	Std	4.877e+02	1.414e+03	8.938e+02	8.335e+02	7.731e+02	4.779e+02	5.451e+02	1.624e+03	1.140e+03
f_{23}	Mean	1.730e+04 +	1.478e+04 +	1.466e+04 +	1.410e+04 +	1.573e+04 +	1.595e+04 +	1.566e+04 +	1.499e+04 +	**1.120e+04**
	Std	4.835e+02	1.212e+03	9.664e+02	1.029e+03	6.755e+02	3.033e+02	5.050e+02	1.412e+03	1.519e+03
f_{24}	Mean	1.668e+03 +	1.437e+03 +	1.437e+03 +	1.396e+03 ≈	1.465e+03 +	1.746e+03 +	1.426e+03 +	1.403e+03 ≈	**1.395e+03**
	Std	1.377e+02	3.196e+01	2.751e+01	1.617e+01	2.673e+01	7.434e+01	1.585e+01	1.938e+01	1.986e+01
f_{25}	Mean	1.593e+03 +	1.542e+03 +	1.548e+03 +	**1.516e+03** −	1.611e+03 +	1.669e+03 +	1.544e+03 +	1.536e+03 ≈	1.529e+03
	Std	2.609e+01	1.657e+01	2.861e+01	1.483e+01	1.706e+01	9.549e+00	1.231e+01	2.402e+01	1.914e+01
f_{26}	Mean	1.740e+03 +	1.695e+03 +	1.690e+03 +	1.667e+03 ≈	1.670e+03 +	1.744e+03 +	1.667e+03 +	**1.653e+03** ≈	1.658e+03
	Std	4.181e+01	1.407e+01	1.010e+01	1.240e+01	4.226e+01	9.082e+01	4.161e+01	4.415e+01	1.267e+01
f_{27}	Mean	4.770e+03 +	3.820e+03 +	3.885e+03 +	3.516e+03 +	3.797e+03 +	5.035e+03 +	3.701e+03 +	3.420e+03 ≈	**3.397e+03**
	Std	5.243e+02	2.129e+02	2.554e+02	1.620e+02	1.373e+02	3.805e+02	8.734e+01	1.585e+02	1.749e+02
f_{28}	Mean	1.332e+04 +	9.646e+03 +	8.483e+03 +	**5.864e+03** −	7.821e+03 +	8.88 +5e+03 +	9.164e+03 +	7.686e+03 +	6.955e+03
	Std	1.573e+03	9.008e+02	8.797e+02	1.129e+03	4.160e+02	3.255e+02	6.379e+02	5.425e+02	4.485e+02
+/≈/−:		28/0/0	26/2/0	26/2/0	20/4/4	27/1/0	27/1/0	27/1/0	23/5/0	–

Table 5 Ablation experiments on 30-D CEC2013 benchmark functions

Func.	SAO		SAO-AUS		SAO$_k$		SAO$_k$-AUS	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	8.346e+03 +	3.755e+03	1.058e+03 +	2.472e+03	1.329e+04 +	4.216e+03	− **1.399e+03**	5.461e+00
f_2	1.233e+08 +	6.047e+07	7.033e+07 +	3.226e+07	1.208e+08 +	5.901e+07	**7.648e+06**	3.566e+06
f_3	7.381e+10 +	3.011e+10	4.017e+10 +	1.844e+10	1.443e+11 +	1.489e+11	**2.114e+09**	2.911e+09
f_4	5.896e+04 +	4.624e+03	5.542e+04 +	6.224e+03	2.566e+04 +	9.272e+03	**8.043e+03**	3.865e+03
f_5	6.559e+03 +	3.667e+03	4.671e+02 +	1.338e+03	7.127e+03 +	4.298e+03	− **9.995e+02**	1.137e+00
f_6	− 2.680e+02 +	2.283e+02	− 6.229e+02 +	9.588e+01	3.099e+02 +	4.672e+02	− **8.043e+02**	3.392e+01
f_7	2.401e+05 +	1.898e+05	1.593e+05 ≈	3.951e+04	1.807e+05 +	6.083e+04	**1.408e+05**	3.819e+04
f_8	− 6.790e+02 ≈	7.552e−02	− **6.790e+02** ≈	9.800e−02	− 6.790e+02 ≈	4.679e−02	− 6.790e+02	5.730e−02
f_9	− 5.688e+02 +	2.498e+00	− 5.694e+02 +	3.014e+00	− 5.704e+02 ≈	2.424e+00	− **5.716e+02**	2.470e+00
f_{10}	8.280e+02 +	5.006e+02	3.265e+02 +	4.620e+02	1.306e+03 +	6.001e+02	− **4.527e+02**	4.658e+01
f_{11}	− 4.651e+01 +	6.067e+01	− 9.614e+01 +	6.646e+01	− 5.149e+01 +	6.477e+01	− **1.436e+02**	5.725e+01
f_{12}	5.160e+01 +	6.123e+01	− 2.204e+01 +	6.302e+01	5.221e+01 +	5.412e+01	− **6.377e+01**	5.238e+01
f_{13}	1.812e+02 ≈	6.348e+01	**1.465e+02** ≈	5.862e+01	1.679e+02 ≈	6.961e+01	1.538e+02	6.121e+01
f_{14}	5.236e+03 +	7.846e+02	4.354e+03 +	7.836e+02	4.823e+03 +	7.228e+02	**3.573e+03**	5.154e+02
f_{15}	5.811e+03 +	5.706e+02	5.523e+03 +	7.079e+02	5.827e+03 +	5.905e+02	**4.113e+03**	6.586e+02
f_{16}	2.015e+02 ≈	4.234e−01	2.014e+02 ≈	3.509e−01	2.013e+02 ≈	6.385e−01	**2.012e+02**	6.767e−01
f_{17}	7.205e+02 +	8.249e+01	6.554e+02 ≈	7.855e+01	6.968e+02 +	6.439e+01	**6.257e+02**	5.677e+01
f_{18}	1.164e+03 +	1.795e+02	1.079e+03 ≈	1.894e+02	1.235e+03 +	1.678e+02	**1.051e+03**	1.510e+02
f_{19}	7.131e+03 +	8.136e+03	2.384e+03 +	1.500e+03	9.868e+03 +	8.082e+03	**1.138e+03**	7.807e+02
f_{20}	6.147e+02 +	3.799e−01	6.147e+02 +	3.813e−01	6.135e+02 ≈	6.051e−01	**6.133e+02**	9.519e−01
f_{21}	2.528e+03 +	1.358e+02	2.229e+03 +	1.502e+02	2.597e+03 +	1.471e+02	**1.912e+03**	1.031e+02
f_{22}	7.491e+03 +	1.446e+03	6.966e+03 +	1.561e+03	6.653e+03 +	7.892e+02	**5.419e+03**	9.909e+02
f_{23}	8.092e+03 +	9.831e+02	7.614e+03 +	1.148e+03	7.029e+03 +	7.610e+02	**5.607e+03**	1.041e+03
f_{24}	1.292e+03 ≈	8.989e+00	**1.288e+03** ≈	1.024e+01	1.289e+03 ≈	9.541e+00	1.290e+03	1.081e+01
f_{25}	**1.414e+03** ≈	1.358e+01	1.415e+03 ≈	1.419e+01	1.414e+03 ≈	1.238e+01	1.415e+03	1.301e+01
f_{26}	1.564e+03 ≈	5.377e+01	**1.564e+03** ≈	5.424e+01	1.570e+03 ≈	4.448e+01	1.566e+03	4.513e+01
f_{27}	2.565e+03 ≈	1.144e+02	2.584e+03 ≈	1.185e+02	**2.555e+03** ≈	9.205e+01	2.558e+03	1.072e+02
f_{28}	4.392e+03 +	2.787e+02	4.366e+03 +	3.566e+02	4.242e+03 ≈	2.887e+02	**4.031e+03**	3.120e+02
+/≈/−	21/7/0		18/10/0		18/10/0		–	

SAO the original snow ablation optimization, *SAO-AUS*: SAO + asynchronous update strategy, *SAO$_k$* SAO + top-k survival mechanism, *SAO$_k$*-AUS SAO + proposed two strategies

strategies termed the AUS, presents a novel selection mechanism in the MA community, particularly in the context of optimization inspired by swarm intelligence. In the original SAO, the utilization of valuable information derived from offspring solutions is often delayed until the subsequent iteration starts. For instance, if an offspring solution, denoted as O_i^t and generated by the individual X_i^t at iteration t, exhibits superior fitness compared to the current optimum solution X_{best}, this information is typically not utilized until the $(t+1)$th iteration. Contrastingly, our proposed AUS revolutionizes this process by promptly incorporating such valuable knowledge to update the current best solution X_{best} promptly. This proactive approach ensures that the optimization process capitalizes on the most recent and relevant information available, leading to expedited convergence towards high-quality solutions.

Moreover, the AUS introduces a dynamic and adaptive component to the MA framework, aligning with the

Table 6 Ablation experiments on 50-D CEC2013 benchmark functions

Function	SAO		SAO-AUS		SAO$_k$		SAO$_k$-AUS	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	2.452e+04 +	6.986e+03	1.013e+04 +	4.534e+03	3.422e+04 +	5.910e+03	**− 5.317e+02**	1.120e+03
f_2	2.458e+08 +	7.497e+07	1.375e+08 +	4.768e+07	2.767e+08 +	1.191e+08	**2.592e+07**	9.483e+06
f_3	1.229e+11 +	5.596e+10	7.607e+10 +	2.017e+10	1.517e+11 +	6.784e+10	**1.340e+10**	1.157e+10
f_4	8.525e+04 +	7.726e+03	8.468e+04 +	6.054e+03	5.223e+04 +	9.847e+03	**2.369e+04**	6.396e+03
f_5	9.344e+03 +	2.790e+03	3.571e+03 +	2.286e+03	1.414e+04 +	5.740e+03	**− 6.996e+02**	3.357e+02
f_6	6.180e+02 +	4.743e+02	− 3.623e+01 +	2.948e+02	1.238e+03 +	6.171e+02	**− 6.602e+02**	6.004e+01
f_7	5.029e+05 +	1.281e+05	5.161e+05 +	2.143e+05	5.820e+05 +	3.777e+05	**4.034e+05**	1.430e+05
f_8	− 6.788e+02 +	5.106e−02	− 6.788e+02 +	6.071e−02	− 6.788e+02 ≈	4.060e−02	**− 6.788e+02**	4.320e−02
f_9	− 5.382e+02 +	3.935e+00	− 5.391e+02 ≈	4.149e+00	− 5.412e+02 ≈	4.742e+00	**− 5.413e+02**	3.868e+00
f_{10}	2.543e+03 +	7.265e+02	1.415e+03 +	5.013e+02	3.595e+03 +	8.860e+02	**− 1.551e+02**	1.408e+02
f_{11}	2.652e+02 +	7.848e+01	1.755e+02 +	7.744e+01	2.797e+02 +	7.013e+01	**8.140e+01**	6.852e+01
f_{12}	4.198e+02 +	6.939e+01	3.364e+02 +	8.573e+01	4.233e+02 +	7.912e+01	**2.142e+02**	8.485e+01
f_{13}	5.192e+02 +	7.126e+01	4.764e+02 ≈	1.011e+02	5.193e+02 +	8.962e+01	**4.494e+02**	7.233e+01
f_{14}	1.076e+04 +	7.836e+02	8.911e+03 +	7.610e+02	1.022e+04 +	6.999e+02	**6.634e+03**	8.540e+02
f_{15}	1.236e+04 +	7.157e+02	1.101e+04 +	9.811e+02	1.210e+04 +	1.051e+03	**9.054e+03**	1.124e+03
f_{16}	2.022e+02 +	5.274e−01	2.022e+02 +	6.062e−01	2.020e+02 ≈	9.176e−01	**2.018e+02**	9.047e−01
f_{17}	1.188e+03 +	9.661e+01	1.101e+03 +	1.181e+02	1.156e+03 +	1.195e+02	**9.652e+02**	9.952e+01
f_{18}	2.122e+03 ≈	2.817e+02	**1.955e+03** ≈	2.167e+02	2.300e+03 +	2.467e+02	1.959e+03	2.640e+02
f_{19}	1.679e+04 +	1.126e+04	5.260e+03 +	2.714e+03	2.567e+04 +	1.766e+04	**2.593e+03**	1.266e+03
f_{20}	6.240e+02 +	6.398e−01	6.240e+02 +	7.179e−01	6.231e+02 ≈	8.055e−01	**6.226e+02**	1.304e+00
f_{21}	3.417e+03 +	2.860e+02	2.646e+03 +	3.207e+02	3.728e+03 +	1.899e+02	**1.354e+03**	1.655e+02
f_{22}	1.350e+04 +	1.624e+03	1.213e+04 +	1.904e+03	1.259e+04 +	9.157e+02	**9.620e+03**	1.140e+03
f_{23}	1.499e+04 +	1.412e+03	1.444e+04 +	1.665e+03	1.367e+04 +	9.185e+02	**1.120e+04**	1.519e+03
f_{24}	1.403e+03 ≈	1.938e+01	1.400e+03 ≈	2.053e+01	1.397e+03 ≈	1.873e+01	**1.395e+03**	1.986e+01
f_{25}	1.536e+03 ≈	2.402e+01	1.533e+03 ≈	1.985e+01	1.534e+03 ≈	1.801e+01	**1.529e+03**	1.914e+01
f_{26}	1.653e+03 ≈	4.415e+01	**1.648e+03** ≈	4.373e+01	1.652e+03 ≈	3.475e+01	1.658e+03	1.267e+01
f_{27}	3.420e+03 ≈	1.585e+02	3.439e+03 ≈	1.356e+02	3.440e+03 ≈	1.447e+02	**3.397e+03**	1.749e+02
f_{28}	7.686e+03 +	5.425e+02	7.372e+03 +	5.546e+02	7.554e+03 +	3.975e+02	**6.955e+03**	4.485e+02
+/≈/−	23/5/0		21/7/0		20/8/0		–	

principles of swarm intelligence. By integrating the evaluation and update process seamlessly into the search operation for every individual, our strategy reflects the timely knowledge-sharing mechanisms observed in natural systems. This mechanism enables individuals within the optimization framework to continuously exchange valuable information, akin to the collaborative behavior exhibited by organisms in nature or agents in a swarm. Consequently, when constructing offspring individuals for subsequent iterations, both search strategies, as defined by Eq. (2) and (4), are designed to utilize the most up-to-date information available, ensuring a more informed exploration and exploitation of the solution space.

The incorporation of the top-k survival mechanism represents a sophisticated selection strategy within the context of MAs, particularly in the optimization landscape governed by the SAO. By selectively retaining solutions with fitness values within the top-k range, this strategy embodies a high-level greed selection approach, ensuring the survival of the best-performing individuals within the

Table 7 The experimental results and statistical analyses on 10-D CEC2020 benchmark functions

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
f_1	Mean	1.320e+10 +	1.990e+07 +	1.209e+08 +	3.147e+07 +	8.284e+08 +	6.862e+09 +	1.389e+09 +	1.514e+08 +	**3.223e+03**
	Std	4.953e+09	1.495e+07	9.334e+07	2.270e+07	6.562e+08	2.254e+09	6.574e+08	2.988e+08	3.172e+03
f_2	Mean	1.168e+12 +	2.676e+09 +	1.427e+10 +	2.583e+09 +	5.439e+10 +	5.475e+11 +	1.387e+11 +	1.009e+10 +	**2.789e+05**
	Std	3.550e+11	3.295e+09	1.690e+10	1.726e+09	2.542e+10	1.836e+11	5.650e+10	1.463e+10	3.477e+05
f_3	Mean	3.917e+11 +	7.483e+08 +	2.934e+09 +	8.586e+08 +	1.170e+10 +	1.464e+11 +	4.112e+10 +	1.549e+09 +	**1.830e+05**
	Std	1.154e+11	7.734e+08	2.533e+09	5.154e+08	1.129e+10	5.057e+10	1.991e+10	4.746e+09	2.277e+05
f_4	Mean	6.628e+04 +	1.913e+03 +	1.910e+03 +	1.906e+03 +	1.928e+03 +	3.167e+03 +	1.954e+03 +	1.903e+03 +	**1.902e+03**
	Std	6.099e+04	9.800e+00	8.122e+00	1.942e+00	4.878e+01	1.313e+03	7.268e+01	1.104e+00	1.816e+00
f_5	Mean	3.328e+06 +	2.400e+04 +	1.576e+04 +	4.099e+04 +	9.071e+04 +	8.150e+04 +	4.480e+04 +	9.426e+04 +	**2.671e+03**
	Std	2.645e+06	2.220e+04	9.738e+03	2.371e+04	9.639e+04	4.311e+04	2.780e+04	1.083e+05	5.946e+02
f_6	Mean	3.235e+05 +	4.314e+03 +	2.523e+03 +	4.682e+03 +	4.829e+03 +	3.372e+03 +	3.707e+03 +	5.982e+03 +	**1.717e+03**
	Std	5.258e+05	4.108e+03	6.663e+02	2.721e+03	3.983e+03	1.132e+03	1.128e+03	3.644e+03	1.696e+02
f_7	Mean	4.007e+06 +	1.828e+04 +	1.218e+04 +	3.790e+04 +	3.653e+04 +	5.635e+04 +	3.382e+04 +	5.901e+04 +	**6.686e+03**
	Std	2.763e+06	1.252e+04	7.362e+03	1.341e+04	4.596e+04	2.412e+04	2.305e+04	1.129e+05	4.776e+03
f_8	Mean	2.389e+03 +	2.315e+03 +	2.313e+03 +	2.311e+03 +	2.315e+03 +	2.332e+03 +	2.324e+03 +	2.308e+03 +	**2.306e+03**
	Std	4.421e+01	4.252e+00	4.371e+00	1.618e+00	3.145e+00	9.634e+00	4.665e+00	2.962e+00	2.312e+00
f_9	Mean	8.033e+03 +	2.824e+03 +	3.006e+03 +	2.823e+03 +	3.333e+03 +	4.721e+03 +	4.418e+03 +	3.110e+03 +	**2.588e+03**
	Std	1.507e+03	2.389e+02	1.907e+02	7.042e+01	4.639e+02	9.472e+02	4.008e+02	4.901e+02	4.807e+01
f_{10}	Mean	3.635e+03 +	3.058e+03 +	3.061e+03 +	3.029e+03 ≈	3.062e+03 +	3.213e+03 +	3.096e+03 +	3.036e+03 +	**3.010e+03**
	Std	2.315e+02	4.450e+01	4.393e+01	3.988e+01	2.111e+01	6.600e+01	3.500e+01	3.804e+01	2.934e+01
+/≈/−:		10/0/0	10/0/0	10/0/0	9/1/0	10/0/0	10/0/0	10/0/0	10/0/0	–

f_1: unimodal function; $f_2 - f_4$: multimodal functions; $f_5 - f_7$: hybrid functions; $f_8 - f_{10}$: composition functions; mean and Std: the mean and the standard deviation of 30 trial runs

population. This targeted preservation of elite solutions plays a pivotal role in the exploration phase of SAO, where individuals within the elite component serve as the foundational vectors for subsequent enhancements. Specifically, the top-*k* survival mechanism is strategically employed to bolster the optimization process by concentrating efforts on refining the second-best and third-best solutions, thus orchestrating a focused and efficient exploration of the solution space. However, the application of the top-*k* survival mechanism warrants careful consideration regarding its potential impact on population diversity, a crucial aspect in MA. According to the proximate optimality principle (POP), which posits that superior solutions often share similar structures [47], the selective retention of elite solutions within the top-*k* range may inadvertently lead to a reduction in population diversity over time. This phe-

Table 8 The experimental results and statistical analyses on 30-D CEC2020 benchmark functions

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
f_1	Mean	7.809e+10 +	1.565e+09 +	5.703e+09 +	7.369e+08 +	2.181e+10 +	4.735e+10 +	3.184e+10 +	1.122e+10 +	**6.692e+06**
	Std	1.110e+10	7.377e+08	2.032e+09	2.392e+08	3.319e+09	6.728e+09	6.500e+09	4.585e+09	2.394e+07
f_2	Mean	9.040e+12 +	1.599e+11 +	7.393e+11 +	7.922e+10 +	2.519e+12 +	6.189e+12 +	3.769e+12 +	1.176e+12 +	**6.099e+08**
	Std	1.263e+12	9.388e+10	2.626e+11	2.458e+10	4.288e+11	8.773e+11	6.981e+11	3.944e+11	2.093e+09
f_3	Mean	2.945e+12 +	5.609e+10 +	2.167e+11 +	2.657e+10 +	6.939e+11 +	1.808e+12 +	1.097e+12 +	4.199e+11 +	**6.921e+07**
	Std	4.425e+11	2.499e+10	6.835e+10	8.762e+09	1.133e+11	2.524e+11	2.081e+11	1.543e+11	1.451e+08
f_4	Mean	3.947e+06 +	4.352e+03 +	8.781e+03 +	**1.944e+03** −	5.893e+04 +	4.809e+05 +	1.609e+05 +	7.747e+03 +	2.390e+03
	Std	1.738e+06	6.719e+03	1.085e+04	2.006e+01	4.171e+04	2.532e+05	1.335e+05	5.234e+03	4.255e+02
f_5	Mean	1.709e+08 +	1.276e+06 +	6.768e+06 +	2.515e+06 +	3.909e+06 +	9.202e+07 +	1.366e+07 +	2.441e+06 +	**7.185e+04**
	Std	6.718e+07	1.120e+06	9.285e+06	1.305e+06	3.142e+06	3.579e+07	6.455e+06	1.829e+06	3.187e+04
f_6	Mean	5.423e+08 +	5.445e+04 +	8.163e+04 +	3.841e+04 +	9.522e+05 +	1.721e+05 +	1.603e+06 +	1.718e+05 +	**5.887e+03**
	Std	3.465e+08	6.806e+04	6.351e+04	1.922e+04	1.196e+06	1.068e+05	1.675e+06	5.032e+05	4.085e+03
f_7	Mean	1.260e+09 +	2.224e+06 +	1.555e+07 +	3.698e+06 +	1.319e+07 +	3.914e+08 +	5.521e+07 +	8.658e+06 +	**1.067e+05**
	Std	7.211e+08	1.227e+06	1.445e+07	2.701e+06	1.302e+07	1.891e+08	3.644e+07	1.152e+07	1.477e+05
f_8	Mean	6.046e+03 +	2.761e+03 +	2.956e+03 +	**2.455e+03** −	2.778e+03 +	3.588e+03 +	3.011e+03 +	2.619e+03 +	2.527e+03
	Std	1.140e+03	2.408e+02	2.942e+02	3.448e+01	9.725e+01	1.745e+02	2.323e+02	8.416e+01	5.979e+01
f_9	Mean	4.320e+04 +	6.918e+03 +	7.646e+03 +	4.573e+03 +	2.129e+04 +	3.213e+04 +	1.554e+04 +	1.298e+04 +	**3.012e+03**
	Std	4.521e+03	2.360e+03	8.538e+02	2.480e+02	3.701e+03	1.491e+03	3.638e+03	3.017e+03	7.220e+02
f_{10}	Mean	1.040e+04 +	3.205e+03 +	3.497e+03 +	3.078e+03 +	4.019e+03 +	6.268e+03 +	4.844e+03 +	3.530e+03 +	**2.973e+03**
	Std	1.987e+03	1.243e+02	1.998e+02	5.297e+01	3.335e+02	7.898e+02	5.896e+02	3.037e+02	2.726e+01
+/≈/−:		10/0/0	10/0/0	10/0/0	8/0/2	10/0/0	10/0/0	10/0/0	10/0/0	–

nomenon raises concerns regarding the exploration-exploitation balance and the risk of premature convergence towards local optima.

Addressing this concern, the dual population mechanism inherent in SAO offers a compelling solution by orchestrating a diverse set of search operators to maintain population diversity throughout the optimization process. By partitioning the population into two distinct subsets-P_a and P_b-the SAO framework ensures that a portion of the population adopts exploitation operators, while the remaining individuals engage in exploration behavior. This dynamic interplay between exploitation and exploration, facilitated by a diverse ensemble of search operators, effectively mitigates the risk of population induced by the top-k survival mechanism.

In essence, the strategic integration of the top-k survival mechanism into the SAO demonstrates a nuanced understanding of optimization dynamics, balancing the imperative for elitism with the necessity of preserving population diversity. By leveraging the dual population mechanism to orchestrate a harmonious interplay between exploration and exploitation, SAO effectively harnesses the benefits of the top-k survival mechanism while safeguarding against the potential pitfalls of diminished population diversity,

Table 9 The experimental results and statistical analyses on 50-D CEC2020 benchmark functions

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
f_1	Mean	1.457e+11 +	5.272e+09 +	1.638e+10 +	2.487e+09 +	6.539e+10 +	1.049e+11 +	8.793e+10 +	4.010e+10 +	**1.886e+09**
	Std	1.879e+10	1.720e+09	5.288e+09	8.201e+08	7.172e+09	8.831e+09	1.275e+10	5.902e+09	2.849e+09
f_2	Mean	1.693e+13 +	5.931e+11 +	1.973e+12 +	2.870e+11 +	6.975e+12 +	1.202e+13 +	9.963e+12 +	4.366e+12 +	**1.603e+11**
	Std	2.701e+12	1.611e+11	4.156e+11	9.457e+10	8.248e+11	1.232e+12	1.461e+12	8.434e+11	1.731e+11
f_3	Mean	5.526e+12 +	2.190e+11 +	6.322e+11 +	9.957e+10 +	2.138e+12 +	4.010e+12 +	3.336e+12 +	1.263e+12 +	**3.524e+10**
	Std	7.666e+11	9.115e+10	1.305e+11	2.476e+10	2.404e+11	3.586e+11	4.116e+11	2.447e+11	4.722e+10
f_4	Mean	1.015e+07 +	1.060e+04 +	3.934e+04 +	**2.384e+03** −	3.246e+05 +	1.809e+06 +	1.250e+06 +	1.063e+05 +	5.124e+03
	Std	4.946e+06	7.589e+03	2.733e+04	4.443e+02	1.541e+05	4.416e+05	5.095e+05	9.874e+04	1.877e+03
f_5	Mean	6.508e+08 +	9.986e+06 +	2.599e+07 +	1.325e+07 +	3.722e+07 +	2.309e+08 +	5.630e+07 +	1.995e+07 +	**6.664e+05**
	Std	2.613e+08	6.492e+06	1.584e+07	5.259e+06	1.710e+07	7.717e+07	2.100e+07	1.070e+07	8.366e+05
f_6	Mean	1.695e+10 +	4.666e+06 +	4.103e+06 +	1.362e+06 +	3.428e+08 +	6.695e+07 +	6.362e+08 +	7.358e+07 +	**1.468e+04**
	Std	8.499e+09	2.130e+07	4.004e+06	7.724e+05	3.793e+08	2.466e+07	4.092e+08	1.875e+08	1.473e+04
f_7	Mean	2.124e+10 +	2.575e+07 +	1.638e+08 +	2.981e+07 +	6.185e+08 +	1.041e+10 +	9.326e+08 +	1.933e+08 +	**6.015e+05**
	Std	9.996e+09	1.952e+07	1.049e+08	1.400e+07	5.408e+08	4.040e+09	4.073e+08	1.794e+08	9.812e+05
f_8	Mean	1.450e+04 +	5.050e+03 +	6.094e+03 +	**2.621e+03** −	6.032e+03 +	9.564e+03 +	5.436e+03 +	4.442e+03 +	3.138e+03
	Std	1.833e+03	1.885e+03	2.021e+03	5.367e+01	1.377e+03	7.212e+02	1.154e+03	1.632e+03	4.131e+02
f_9	Mean	7.932e+04 +	1.504e+04 +	1.893e+04 +	**6.088e+03** −	5.194e+04 +	6.315e+04 +	5.487e+04 +	3.672e+04 +	1.136e+04
	Std	8.280e+03	6.505e+03	7.827e+03	9.458e+02	3.256e+03	1.827e+03	8.188e+03	6.921e+03	6.860e+03
f_{10}	Mean	4.364e+04 +	4.665e+03 +	7.100e+03 +	4.237e+03 +	1.203e+04 +	2.102e+04 +	1.581e+04 +	8.263e+03 +	**3.745e+03**
	Std	1.008e+04	4.812e+02	9.033e+02	2.707e+02	1.661e+03	2.660e+03	2.412e+03	1.381e+03	2.283e+02
+/≈/−:		10/0/0	10/0/0	10/0/0	7/0/3	10/0/0	10/0/0	10/0/0	10/0/0	–

thereby enhancing the robustness and efficacy of the optimization process.

Expanding on the potential of the proposed strategies beyond the scope of SAO opens up a promising avenue for advancing population-based MAs. The AUS and the top-k survival mechanism exhibit versatility and applicability across a range of MAs, including DE, GWO, and WOA. By seamlessly integrating these strategies into existing MAs, researchers can unlock new opportunities for enhancing their performance and addressing complex optimization challenges.

Moreover, since these strategies have been designed independently, they offer the flexibility to be used individually or in combination to suit the specific characteristics of the optimization problem at hand. This opens up a rich landscape for further research into the dynamic and intelligent selection of strategy combinations based on problem characteristics, such as problem dimensionality, search space topology, and solution landscape complexity. By intelligently selecting and combining these strategies, researchers can tailor MAs to effectively tackle a wide range of optimization challenges, from high-dimensional

Table 10 The experimental results and statistical analyses on 100-D CEC2020 benchmark functions

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
f_1	Mean	2.998e+11 +	1.946e+10 −	5.291e+10 +	**1.042e+10** −	1.981e+11 +	2.406e+11 +	2.458e+11 +	1.456e+11 +	2.544e+10
	Std	1.838e+10	4.094e+09	9.944e+09	1.540e+09	8.927e+09	4.621e+09	1.624e+10	1.151e+10	8.634e+09
f_2	Mean	3.556e+13 +	2.166e+12 −	5.364e+12 +	**1.115e+12** −	2.199e+13 +	2.747e+13 +	2.767e+13 +	1.592e+13 +	3.057e+12
	Std	2.548e+12	5.190e+11	8.395e+11	2.311e+11	1.146e+12	6.529e+11	1.621e+12	1.541e+12	1.078e+12
f_3	Mean	1.213e+13 +	7.185e+11 ≈	1.969e+12 +	**3.922e+11** −	6.936e+12 +	8.794e+12 +	9.494e+12 +	5.471e+12 +	9.217e+11
	Std	7.716e+11	1.671e+11	3.111e+11	5.386e+10	3.872e+11	9.461e+10	8.348e+11	6.147e+11	3.629e+11
f_4	Mean	1.259e+07 +	3.352e+04 +	8.178e+04 +	**5.183e+03** −	1.214e+06 +	4.876e+06 +	4.932e+06 +	5.570e+05 +	1.653e+04
	Std	2.919e+06	2.152e+04	5.197e+04	1.906e+03	3.457e+05	1.043e+06	1.517e+06	2.151e+05	7.149e+03
f_5	Mean	2.791e+09 +	7.322e+07 +	1.371e+08 +	7.965e+07 +	2.649e+08 +	7.063e+08 +	5.000e+08 +	1.980e+08 +	**1.028e+07**
	Std	6.591e+08	2.437e+07	4.729e+07	2.088e+07	4.709e+07	1.214e+08	1.372e+08	6.054e+07	5.110e+06
f_6	Mean	8.046e+10 +	1.041e+07 +	4.892e+07 +	9.042e+05 +	5.419e+09 +	3.801e+10 +	1.169e+10 +	1.916e+09 +	**5.623e+05**
	Std	2.181e+10	2.847e+07	4.717e+07	5.364e+05	2.567e+09	8.679e+09	4.728e+09	1.223e+09	6.849e+05
f_7	Mean	5.045e+10 +	1.210e+08 +	6.961e+08 +	1.277e+08 +	5.782e+09 +	2.342e+10 +	7.907e+09 +	2.432e+09 +	**7.180e+07**
	Std	9.995e+09	7.343e+07	2.676e+08	3.950e+07	2.240e+09	4.677e+09	2.940e+09	2.155e+09	1.042e+08
f_8	Mean	3.113e+04 +	1.446e+04 +	1.757e+04 +	**2.838e+03** −	1.685e+04 +	2.267e+04 +	1.687e+04 +	1.205e+04 +	6.124e+03
	Std	1.461e+03	5.170e+03	4.633e+03	1.023e+02	2.645e+03	1.113e+03	4.034e+03	3.786e+03	2.088e+03
f_9	Mean	2.059e+05 +	6.685e+04 ≈	8.252e+04 +	**2.507e+04** −	1.504e+05 +	1.609e+05 +	1.794e+05 +	1.350e+05 +	6.471e+04
	Std	1.348e+04	3.570e+04	2.163e+04	5.812e+03	4.940e+03	7.001e+02	4.601e+03	8.462e+03	2.056e+04
f_{10}	Mean	5.463e+04 +	5.458e+03 ≈	8.191e+03 +	**4.825e+03** −	1.924e+04 +	3.885e+04 +	3.163e+04 +	1.436e+04 +	5.573e+03
	Std	8.763e+03	4.240e+02	9.103e+02	2.736e+02	1.857e+03	4.177e+03	4.829e+03	2.182e+03	5.059e+02
+/≈/−:		10/0/0	5/3/2	10/0/0	3/0/7	10/0/0	10/0/0	10/0/0	10/0/0	–

and multimodal problems to constrained and dynamic environments.

5.3 Performance analysis on CEC2013 and CEC2020 benchmark functions

Since CEC2013 and CEC2020 suites contain 28 and 10 benchmark functions respectively, each exhibiting a variety of characteristics in their fitness landscapes. These benchmarks serve as valuable tools for evaluating the overall optimization performance of algorithms.

In our comprehensive numerical experiments, Tables 1, 2, 3, and 4 summarize the optimization results on CEC2013 benchmark functions, and Tables 7, 8, 9, and 10 list the experimental results and statistical analyses on CEC2020 benchmark functions. In comparison with eight state-of-the-art optimization algorithms, our proposed SAO$_k$-AUS exhibits competitiveness and, on most benchmark functions, surpasses its competitor algorithms significantly. These comprehensive numerical experiments are sufficient to prove the effectiveness and efficiency of SAO$_k$-AUS practically. Besides, when compared to the

Table 11 The experimental results and statistical analyses on engineering problems

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
WBP	Mean	2.383e+00 +	2.262e+00 +	2.028e+00 +	1.809e+00 +	2.002e+00 +	2.325e+00 +	1.838e+00 +	1.910e+00 +	**1.759e+00**
	Std	3.481e−01	5.372e−01	3.771e−01	1.928e−01	2.044e−01	1.598e−01	8.531e−02	2.551e−01	8.826e−02
	Best	1.889e+00	1.722e+00	1.717e+00	1.688e+00	1.725e+00	1.967e+00	1.698e+00	1.682e+00	1.682e+00
	Worst	3.188e+00	3.841e+00	3.603e+00	2.626e+00	2.484e+00	2.638e+00	2.024e+00	2.894e+00	2.074e+00
PVP	Mean	5.777e+04 +	9.220e+03 +	8.285e+03 +	6.743e+03 +	7.850e+03 +	2.962e+03 −	**2.923e+03** −	5.876e+03 ≈	5.975e+03
	Std	7.765e+04	3.818e+03	2.734e+03	2.452e+03	2.225e+03	4.110e+01	7.363e+01	2.780e+03	2.565e+03
	Best	9.364e+03	2.892e+03	2.892e+03	2.903e+03	2.892e+03	2.916e+03	2.892e+03	2.892e+03	2.892e+03
	Worst	2.717e+05	1.767e+04	1.366e+04	8.383e+03	1.015e+04	3.160e+03	3.282e+03	9.549e+03	8.333e+03
CSP	Mean	6.515e−03 +	6.099e−03 +	6.076e−03 +	6.076e−03 +	6.076e−03 +	6.076e−03 +	6.076e−03 +	6.101e−03 +	**6.076e−03**
	Std	5.299e−04	1.230e−04	1.900e−18	3.899e−07	1.697e−07	6.272e−07	7.065e−09	1.220e−04	0.000e+00
	Best	6.077e−03	6.076e−03	6.076e−03	6.076e−03	6.076e−03	6.076e−03	6.076e−03	6.076e−03	6.076e−03
	Worst	8.540e−03	6.761e−03	6.076e−03	6.078e−03	6.077e−03	6.078e−03	6.076e−03	6.758e−03	6.076e−03
SRD	Mean	3.247e+03 +	3.074e+03 +	3.041e+03 +	2.996e+03 +	2.295e+05 +	3.057e+03 +	3.012e+03 +	2.999e+03 +	**2.987e+03**
	Std	4.327e+01	9.448e+01	6.263e+01	5.811e+00	9.689e+05	1.100e+01	1.213e+01	9.112e+00	1.641e+00
	Best	3.168e+03	2.987e+03	2.989e+03	2.988e+03	2.987e+03	3.034e+03	2.993e+03	2.988e+03	2.986e+03
	Worst	3.363e+03	3.363e+03	3.285e+03	3.011e+03	5.356e+06	3.076e+03	3.046e+03	3.037e+03	2.996e+03
TBTD	Mean	2.686e+02 +	2.641e+02 +	**2.639e+02** −	2.639e+02 +	2.644e+02 +	2.639e+02 +	2.639e+02 ≈	2.642e+02 +	2.639e+02
	Std	2.099e+00	3.837e−01	4.977e−04	5.208e−02	6.955e−01	2.007e−02	5.999e−03	1.107e+00	5.632e−03
	Best	2.641e+02	2.639e+02	2.639e+02	2.639e+02	2.639e+02	2.639e+02	2.639e+02	2.639e+02	2.639e+02
	Worst	2.710e+02	2.652e+02	2.639e+02	2.640e+02	2.669e+02	2.639e+02	2.639e+02	2.698e+02	2.639e+02
GTD	Mean	1.180e−05 +	**0.000e+00** −	**0.000e+00** −	2.052e−11 +	1.076e−08 +	1.474e−11 +	1.557e−15 −	**0.000e+00** −	2.506e−13
	Std	2.748e−05	0.000e+00	0.000e+00	3.680e−11	2.610e−08	3.115e−11	8.387e−15	0.000e+00	1.798e−13
	Best	3.905e−11	0.000e+00	0.000e+00	1.899e−16	4.888e−17	3.209e−15	0.000e+00	0.000e+00	5.518e−23
	Worst	1.312e−04	0.000e+00	0.000e+00	1.479e−10	9.968e−08	1.764e−10	4.672e−14	0.000e+00	7.354e−13

Mean and std: the mean and the standard deviation of 30 trial runs; best and worst: the best and the worst final solution found among 30 trial runs

original SAO on CEC2013 and CEC2020 benchmark functions, we do not observe significant deterioration, which underscores the success of our two proposed strategies.

The previously mentioned examples underscore the remarkable optimization prowess demonstrated by SAO*k*-AUS. Nevertheless, instances of performance degradation have been observed in certain scenarios, such as with function *f*25 in the CEC2013 benchmark between TSO and SAO$_k$-AUS. Moreover, this degradation becomes more pronounced as the dimensionality of the optimization problems increases, particularly evident in the CEC2020 benchmark functions. With increasing dimensions, the competitiveness of SAO$_k$-AUS decreases, indicating its potential limitations in effectively addressing these specific challenges. It is worth noting that while SAO$_k$-AUS may not excel in handling these particular problems, its advancement over the original SAO suggests that the observed degeneration could be attributed to the inherent search strategies of the original SAO. These strategies may struggle to construct high-quality solutions for complex problems and expedite optimization convergence in such contexts, thus contributing to the observed performance deterioration.

5.4 Performance analysis on ablation experiments

In our ablation experiments, we conducted a comparison among the original SAO, SAO-AUS (SAO + asynchronous update strategy), SAO$_k$ (SAO + top-*k* survival mechanism), and SAO$_k$-AUS (SAO + both proposed

Table 12 The experimental results and statistical analyses on engineering problems

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
CBD	Mean	4.083e+00 +	2.815e+00 +	1.527e+00 +	1.350e+00 +	1.384e+00 +	1.344e+00 +	1.546e+00 +	1.341e+00 +	**1.340e+00**
	Std	1.891e+00	1.125e+00	1.211e−01	6.854e−03	2.772e−02	4.126e−03	1.634e−01	1.785e−03	4.711e−05
	Best	1.579e+00	1.422e+00	1.353e+00	1.341e+00	1.345e+00	1.341e+00	1.354e+00	1.341e+00	1.340e+00
	Worst	8.355e+00	5.537e+00	1.850e+00	1.368e+00	1.459e+00	1.365e+00	2.083e+00	1.348e+00	1.340e+00
IBD	Mean	2.302e−04 +	1.746e−04 +	1.745e−04 ≈	1.745e−04 ≈	1.774e−04 +	1.745e−04 +	1.745e−04 +	1.745e−04 +	**1.745e−04**
	Std	6.264e−05	3.790e−07	1.481e−18	4.871e−11	2.440e−06	5.618e−11	4.460e−12	4.574e−12	1.355e−19
	Best	1.746e−04	1.745e−04	1.745e−04	1.746e−04	1.749e−04	1.745e−04	1.745e−04	1.745e−04	1.745e−04
	Worst	4.168e−04	1.761e−04	1.745e−04	1.745e−04	1.840e−04	1.745e−04	1.745e−04	1.745e−04	1.745e−04
TCD	Mean	3.090e+01 +	3.025e+01 +	3.016e+01 ≈	3.017e+01 +	3.027e+01 +	3.016e+01 +	3.016e+01 +	3.016e+01 +	**3.015e+01**
	Std	4.425e−01	3.883e−01	5.085e−11	1.837e−02	1.115e−01	2.641e−02	8.186e−03	1.056e−02	1.065e−14
	Best	3.028e+01	3.015e+01	3.015e+01	3.016e+01	3.016e+01	3.015e+01	3.015e+01	3.015e+01	3.015e+01
	Worst	3.211e+01	3.221e+01	3.016e+01	3.022e+01	3.062e+01	3.028e+01	3.018e+01	3.020e+01	3.015e+01
PLD	Mean	2.829e+02 +	1.923e+02 +	1.407e+02 +	6.656e+00 ≈	2.744e+01 +	7.188e+00 +	**4.099e+00** −	1.700e+01 +	6.604e+00
	Std	2.916e+02	1.660e+02	1.296e+02	3.006e+01	7.978e+01	2.818e+00	3.580e+00	4.673e+01	2.987e+01
	Best	5.350e+00	1.057e+00	1.057e+00	1.060e+00	1.062e+00	3.524e+00	1.079e+00	1.057e+00	1.057e+00
	Worst	1.175e+03	5.603e+02	4.674e+02	1.682e+02	3.193e+02	1.547e+01	1.588e+01	1.720e+02	1.674e+02
CBHD	Mean	8.514e+00 +	7.250e+00 +	6.995e+00 +	7.031e+00 +	7.724e+00 +	7.070e+00 +	7.055e+00 +	6.939e+00 +	**6.843e+00**
	Std	6.649e−01	4.021e−01	1.540e−01	1.174e−01	5.454e−01	6.131e−02	1.128e−01	1.530e−01	2.895e−08
	Best	7.722e+00	6.877e+00	6.844e+00	6.883e+00	7.059e+00	6.964e+00	6.891e+00	6.843e+00	6.843e+00
	Worst	1.006e+01	8.471e+00	7.509e+00	7.407e+00	9.360e+00	7.198e+00	7.429e+00	7.510e+00	6.843e+00
RCB	Mean	1.622e+02 −	1.653e+02 −	1.663e+02 ≈	**1.617e+02** −	1.681e+02 +	1.670e+02 ≈	1.654e+02 −	1.672e+02 ≈	1.669e+02
	Std	2.370e+00	2.649e+00	1.379e+00	2.356e+00	2.023e+00	7.940e−01	7.146e−01	1.927e+00	9.556e−01
	Best	1.594e+02	1.594e+02	1.597e+02	1.594e+02	1.635e+02	1.635e+02	1.635e+02	1.645e+02	1.635e+02
	Worst	1.676e+02	1.682e+02	1.667e+02	1.667e+02	1.741e+02	1.683e+02	1.669e+02	1.7522e+02	1.696e+02
		+/≈/−: 11/0/1	+/≈/−: 10/0/2	+/≈/−: 7/2/3	+/≈/−: 9/2/1	+/≈/−: 12/0/0	+/≈/−: 11/1/0	+/≈/−: 7/2/3	+/≈/−: 10/1/1	

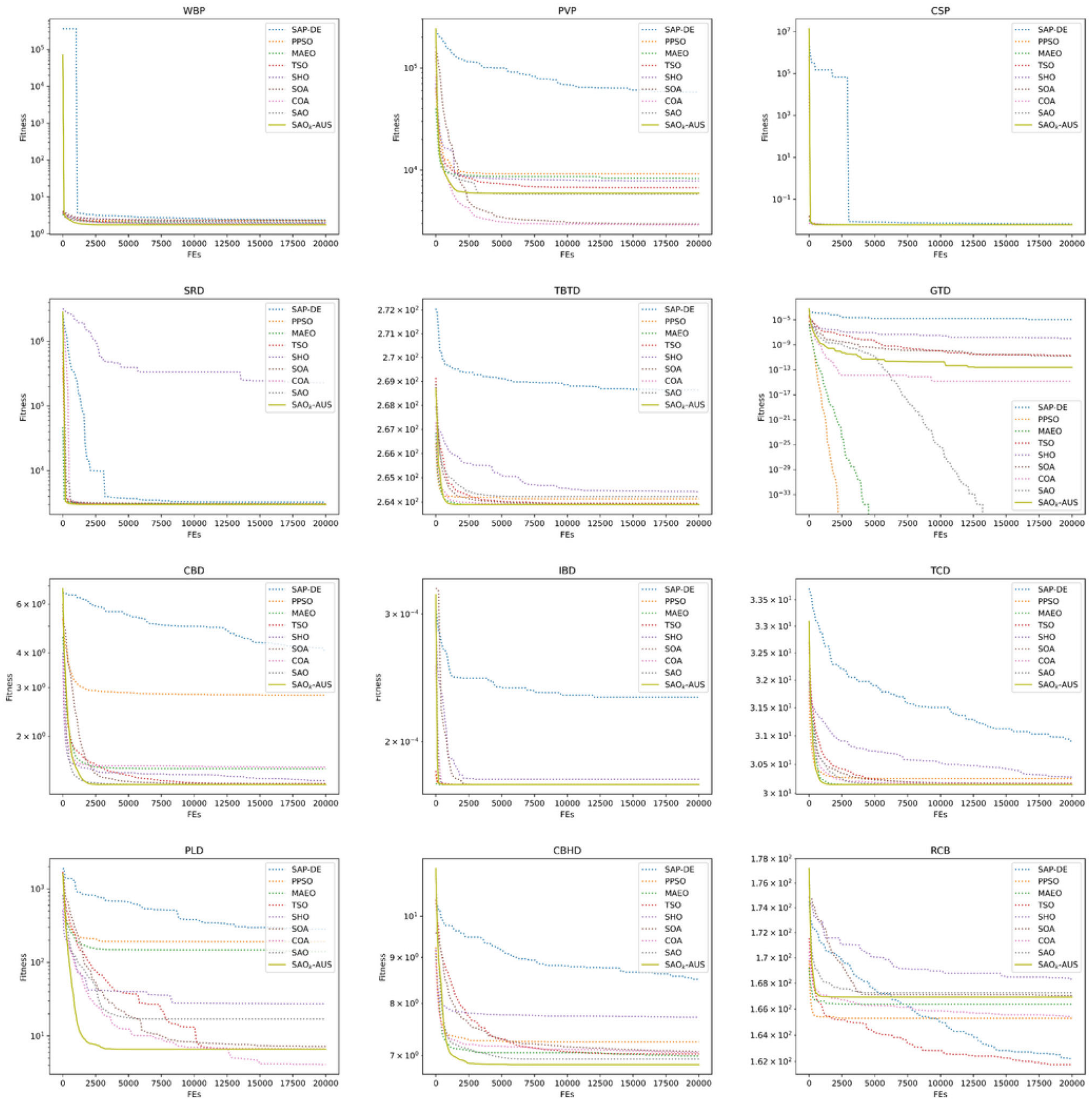


Fig. 3 Convergence curves of optimization algorithms on 12 engineering optimization problems

strategies) on 30-D and 50-D CEC2013 benchmark functions to investigate the individual contributions of each component. The experimental results and statistical analyses presented in Tables 5 and 6 demonstrate the effectiveness of both approaches in enhancing the performance of the original SAO. In addition, we observed an interesting phenomenon: for f_1, the improvement achieved by a single strategy is not apparent, with SAO$_k$ even performing worse than the original SAO in both 30-D and 50-D, but the combination of both approaches exhibits a significant advantage over the original SAO. This pattern is also evident in other functions, such as f_5, f_6, and f_{10}. Therefore, we recommend hybridizing these two proposed approaches to improve the performance of the original SAO.

5.5 Performance analysis on engineering optimization problems

Engineering problems are crucial benchmarks for evaluating the efficacy of MAs in tackling real-world challenges.

Table 13 The summary of the mean of fitness, *PSNR*, and *SSIM* within 30 trial runs when number of threshold (*nThre*) is 4

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
img-1	Fitness	2.8439e+03	2.8825e+03	**2.8844e+03**	2.8829e+03	2.8538e+03	2.8806e+03	2.8823e+03	2.8785e+03	2.8835e+03
	PSNR	1.8278e+01	1.8996e+01	**2.1351e+01**	1.9067e+01	1.8951e+01	1.8802e+01	1.8949e+01	1.9369e+01	1.9367e+01
	SSIM	5.7796e−01	6.0968e−01	**6.4534e−01**	6.1121e−01	6.0569e−01	6.0529e−01	6.0865e−01	6.1632e−01	6.1570e−01
img-2	Fitness	6.0433e+02	6.2446e+02	6.2590e+02	6.2602e+02	6.0921e+02	6.2569e+02	**6.2647e+02**	6.2489e+02	6.2589e+02
	PSNR	1.6211e+01	1.6111e+01	1.6071e+01	1.6216e+01	1.5972e+01	1.6333e+01	**1.6414e+01**	1.5895e+01	1.6050e+01
	SSIM	7.3351e−01	6.9983e−01	**7.3963e−01**	7.0277e−01	7.0543e−01	7.0600e−01	7.0340e−01	7.1124e−01	7.0341e−01
img-3	Fitness	1.8720e+03	1.8890e+03	**1.8923e+03**	1.8915e+03	1.8784e+03	1.8913e+03	1.8916e+03	1.8916e+03	1.8903e+03
	PSNR	1.4997e+01	1.4501e+01	**1.5145e+01**	1.4857e+01	1.4768e+01	1.5024e+01	1.4974e+01	1.50555e+01	1.5125e+01
	SSIM	7.7886e−01	7.9320e−01	**8.1877e−01**	7.9152e−01	7.9291e−01	7.7887e−01	7.8395e−01	7.9904e−01	8.0452e−01
img-4	Fitness	2.7311e+03	2.7488e+03	2.7498e+03	2.7489e+03	2.7355e+03	2.7482e+03	2.7491e+03	2.7492e+03	**2.7499e+03**
	PSNR	2.1067e+01	2.2359e+01	2.2405e+01	2.2433e+01	2.2376e+01	2.2153e+01	2.2442e+01	2.2460e+01	**2.2465e+01**
	SSIM	8.3589e−01	8.5500e−01	8.5189e−01	8.5673e−01	8.5616e−01	8.5104e−01	8.5648e−01	8.5377e−01	**8.5674e−01**
img-5	Fitness	2.9648e+03	2.9967e+03	**2.9992e+03**	2.9973e+03	2.9735e+03	2.9955e+03	2.9970e+03	2.9964e+03	2.9982e+03
	PSNR	1.8841e+01	1.9365e+01	**2.2277e+01**	1.8860e+01	1.9050e+01	1.8246e+01	1.8846e+01	1.9099e+01	1.9201e+01
	SSIM	6.4989e−01	6.6680e−01	**7.0077e−01**	6.6377e−01	6.6146e−01	6.5227e−01	6.6169e−01	6.6255e−01	6.6642e−01
img-6	Fitness	1.6337e+03	1.6674e+03	**1.6695e+03**	1.6681e+03	1.6491e+03	1.6674e+03	1.6685e+03	1.6689e+03	1.6686e+03
	PSNR	2.0225e+01	2.0631e+01	**2.0941e+01**	2.0500e+01	2.0395e+01	2.0351e+01	2.0458e+01	2.0331e+01	2.0403e+01
	SSIM	5.9549e−01	6.0850e−01	**6.2069e−01**	6.1014e−01	6.0761e−01	6.0246e−01	6.0830e−01	6.1166e−01	6.0795e−01
img-7	Fitness	1.6328e+03	1.6711e+03	**1.6755e+03**	1.6737e+03	1.6461e+03	1.6721e+03	1.6744e+03	1.6753e+03	1.6753e+03
	PSNR	1.7618e+01	1.8773e+01	1.8725e+01	1.8953e+01	1.8768e+01	1.8860e+01	1.8955e+01	**1.8967e+01**	1.8933e+01
	SSIM	5.7451e−01	6.0297e−01	**6.4559e−01**	6.1100e−01	6.0687e−01	6.0462e−01	6.0985e−01	6.1279e−01	6.1037e−01
img-8	Fitness	1.3371e+03	1.3572e+03	**1.3583e+03**	1.3573e+03	1.3397e+03	1.3558e+03	1.3571e+03	1.3566e+03	1.3576e+03
	PSNR	1.8742e+01	1.8987e+01	**2.2157e+01**	1.8938e+01	1.8988e+01	1.8344e+01	1.8762e+01	1.9176e+01	1.8910e+01
	SSIM	7.4958e−01	7.8072e−01	**8.1111e−01**	7.8293e−01	7.8317e−01	7.5882e−01	7.7355e−01	7.8540e−01	7.7892e−01

Table 14 The summary of the mean of fitness, *PSNR*, and *SSIM* within 30 trial runs when number of threshold (*nThre*) is 8

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
img-1	Fitness	2.9631e+03	2.9840e+03	**2.9882e+03**	2.9847e+03	2.9599e+03	2.9833e+03	2.9826e+03	2.9854e+03	2.9856e+03
	PSNR	2.1087e+01	2.1658e+01	**2.3090e+01**	2.2128e+01	2.1766e+01	2.1022e+01	2.2620e+01	2.2782e+01	2.2491e+01
	SSIM	7.3359e−01	7.4310e−01	**8.0715e−01**	7.4901e−01	7.3945e−01	7.3297e−01	7.5707e−01	7.6314e−01	7.4294e−01
img-2	Fitness	6.3532e+02	6.4261e+02	6.4563e+02	6.4593e+02	6.3513e+02	6.4569e+02	6.4514e+02	6.4629e+02	**6.4656e+02**
	PSNR	1.4484e+01	1.4768e+01	1.4753e+01	1.5342e+01	1.4952e+01	1.5392e+01	1.5053e+01	1.5184e+01	**1.5628e+01**
	SSIM	6.3391e−01	7.0013e−01	7.0350e−01	6.7655e−01	6.8191e−01	6.6961e−01	6.7764e−01	6.9168e−01	**7.2206e−01**
img-3	Fitness	1.9054e+03	1.9147e+03	1.9177e+03	1.9193e+03	1.9093e+03	1.9187e+03	1.9183e+03	1.9185e+03	**1.9202e+03**
	PSNR	1.3130e+01	1.3065e+01	1.3444e+01	1.3817e+01	1.2889e+01	1.3801e+01	1.3914e+01	1.3820e+01	**1.4021e+01**
	SSIM	7.7591e−01	7.6253e−01	7.7140e−01	**7.9225e−01**	7.7615e−01	7.8528e−01	7.7499e−01	7.8842e−01	7.7606e−01
img-4	Fitness	2.7764e+03	2.7869e+03	2.7892e+03	2.7878e+03	2.7765e+03	2.7873e+03	2.7874e+03	2.7882e+03	**2.7893e+03**
	PSNR	2.2092e+01	2.1648e+01	2.1615e+01	2.1749e+01	2.1662e+01	2.2569e+01	2.1639e+01	2.2656e+01	**2.2912e+01**
	SSIM	7.3156e−01	7.8872e−01	7.8784e−01	7.8134e−01	7.8755e−01	**7.9747e−01**	7.8091e−01	7.9586e−01	7.9121e−01
img-5	Fitness	3.0787e+03	3.0987e+03	3.1025e+03	3.0997e+03	3.0793e+03	3.0988e+03	3.0977e+03	3.1005e+03	**3.1026e+03**
	PSNR	2.1020e+01	2.1302e+01	2.1268e+01	2.2982e+01	**2.3263e+01**	2.0853e+01	2.2623e+01	2.2981e+01	2.2475e+01
	SSIM	7.7262e−01	7.8259e−01	7.8813e−01	7.9878e−01	7.9731e−01	7.8114e−01	7.9195e−01	8.0371e−01	**8.1053e−01**
img-6	Fitness	1.7179e+03	1.7341e+03	1.7389e+03	1.7386e+03	1.7211e+03	1.7366e+03	1.7363e+03	1.7389e+03	**1.7398e+03**
	PSNR	2.1100e+01	2.1417e+01	2.2130e+01	**2.3179e+01**	2.2054e+01	2.2223e+01	2.2321e+01	2.1238e+01	2.2407e+01
	SSIM	6.9252e−01	7.1398e−01	6.7975e−01	7.1775e−01	7.1242e−01	7.0207e−01	7.1846e−01	7.1459e−01	**7.1935e−01**
img-7	Fitness	1.7064e+03	1.7212e+03	**1.7258e+03**	1.7242e+03	1.7041e+03	1.7239e+03	1.7227e+03	1.7245e+03	1.7247e+03
	PSNR	1.9260e+01	1.9169e+01	1.9022e+01	1.9515e+01	1.9022e+01	**1.9607e+01**	1.9404e+01	1.9085e+01	1.9124e+01
	SSIM	6.9026e−01	6.6945e−01	**7.3648e−01**	6.9715e−01	6.8607e−01	6.8017e−01	6.9555e−01	7.2385e−01	6.9430e−01
img-8	Fitness	1.3824e+03	1.3946e+03	**1.3968e+03**	1.3961e+03	1.3808e+03	1.3957e+03	1.3954e+03	1.3964e+03	1.3958e+03
	PSNR	1.8754e+01	1.9020e+01	**2.1418e+01**	1.7963e+01	1.7361e+01	1.6161e+01	1.7463e+01	1.8522e+01	1.9227e+01
	SSIM	7.4530e−01	7.4737e−01	**8.4986e−01**	7.5500e−01	7.2847e−01	7.0216e−01	7.2473e−01	7.5645e−01	7.6572e−01

Table 15 The summary of the mean of fitness, *PSNR*, and *SSIM* within 30 trial runs when number of threshold (*nThre*) is 12

Func.		SAP-DE	PPSO	MAEO	TSO	SHO	SOA	COA	SAO	SAO$_k$-AUS
img-1	Fitness	2.9960e+03	3.0090e+03	3.0112e+03	3.0104e+03	2.9952e+03	3.0087e+03	3.0079e+03	3.0107e+03	**3.0122e+03**
	PSNR	2.3547e+01	2.2406e+01	2.2805e+01	2.2803e+01	2.2860e+01	2.1239e+01	2.3090e+01	2.2740e+01	**2.3359e+01**
	SSIM	8.1771e−01	7.9809e−01	7.9412e−01	8.0307e−01	7.9583e−01	7.7676e−01	8.0942e−01	8.0630e−01	**8.1094e−01**
img-2	Fitness	6.4417e+02	6.5002e+02	6.5142e+02	6.5173e+02	6.4529e+02	6.5186e+02	6.5108e+02	6.5131e+02	**6.5208e+02**
	PSNR	1.6486e+01	1.6040e+01	1.6363e+01	1.6030e+01	1.5649e+01	1.6018e+01	1.6144e+01	1.5852e+01	**1.6807e+01**
	SSIM	7.0190e−01	7.0285e−01	**7.2767e−01**	7.1989e−01	7.1389e−01	7.2313e−01	7.0684e−01	7.2546e−01	7.2074e−01
img-3	Fitness	1.9173e+03	1.9246e+03	1.9263e+03	1.9263e+03	1.9205e+03	1.9265e+03	1.9262e+03	1.9268e+03	**1.9270e+03**
	PSNR	1.4169e+01	1.4024e+01	1.4602e+01	1.5069e+01	1.3684e+01	1.5221e+01	1.4312e+01	1.3541e+01	**1.5351e+01**
	SSIM	7.5886e−01	7.3757e−01	7.5751e−01	8.0221e−01	7.9226e−01	8.0413e−01	7.9332e−01	7.9790e−01	**8.0644e−01**
img-4	Fitness	2.7913e+03	2.7991e+03	**2.8003e+03**	2.7995e+03	2.7906e+03	2.7989e+03	2.7992e+03	2.8000e+03	2.7998e+03
	PSNR	2.2412e+01	2.2574e+01	2.2787e+01	2.2206e+01	2.2364e+01	**2.2904e+01**	2.1853e+01	2.1981e+01	2.1716e+01
	SSIM	8.3273e−01	8.2054e−01	**8.5235e−01**	8.1544e−01	8.1084e−01	8.2383e−01	8.0370e−01	8.2061e−01	8.2322e−01
img-5	Fitness	3.1129e+03	3.1261e+03	3.1284e+03	3.1285e+03	3.1122e+03	3.1265e+03	3.1264e+03	3.1273e+03	**3.1290e+03**
	PSNR	2.1854e+01	2.3759e+01	2.2337e+01	2.3231e+01	2.3606e+01	2.1339e+01	2.3558e+01	2.3705e+01	**2.3787e+01**
	SSIM	8.2232e−01	8.4987e−01	8.4960e−01	8.5403e−01	8.5098e−01	8.3471e−01	8.4997e−01	8.5078e−01	**8.5862e−01**
img-6	Fitness	1.7439e+03	1.7536e+03	1.7571e+03	1.7572e+03	1.7456e+03	1.7563e+03	1.7554e+03	**1.7577e+03**	1.7576e+03
	PSNR	2.2332e+01	2.2939e+01	2.3042e+01	2.2613e+01	2.1439e+01	**2.2861e+01**	2.1307e+01	2.1299e+01	2.2437e+01
	SSIM	7.5660e−01	7.6111e−01	7.2253e−01	7.5144e−01	7.3846e−01	7.4485e−01	7.5162e−01	**7.5813e−01**	7.5684e−01
img-7	Fitness	1.7263e+03	1.7371e+03	1.7395e+03	1.7388e+03	1.7268e+03	1.7384e+03	1.7377e+03	1.7390e+03	**1.7396e+03**
	PSNR	1.9156e+01	1.9352e+01	1.9143e+01	1.9204e+01	1.9012e+01	1.9131e+01	1.9346e+01	1.9002e+01	**1.9852e+01**
	SSIM	7.5813e−01	7.4555e−01	7.6077e−01	7.3636e−01	7.3946e−01	7.2796e−01	7.5030e−01	7.6904e−01	**7.7267e−01**
img-8	Fitness	1.3993e+03	1.4066e+03	1.4083e+03	1.4082e+03	1.3972e+03	1.4076e+03	1.4073e+03	1.4083e+03	**1.4090e+03**
	PSNR	1.9530e+01	1.9598e+01	1.9446e+01	1.9484e+01	1.8779e+01	1.8173e+01	1.8948e+01	1.9512e+01	**1.9657e+01**
	SSIM	8.0164e−01	**8.0178e−01**	7.7173e−01	7.8599e−01	7.5467e−01	7.4087e−01	7.6980e−01	7.6999e−01	7.7741e−01

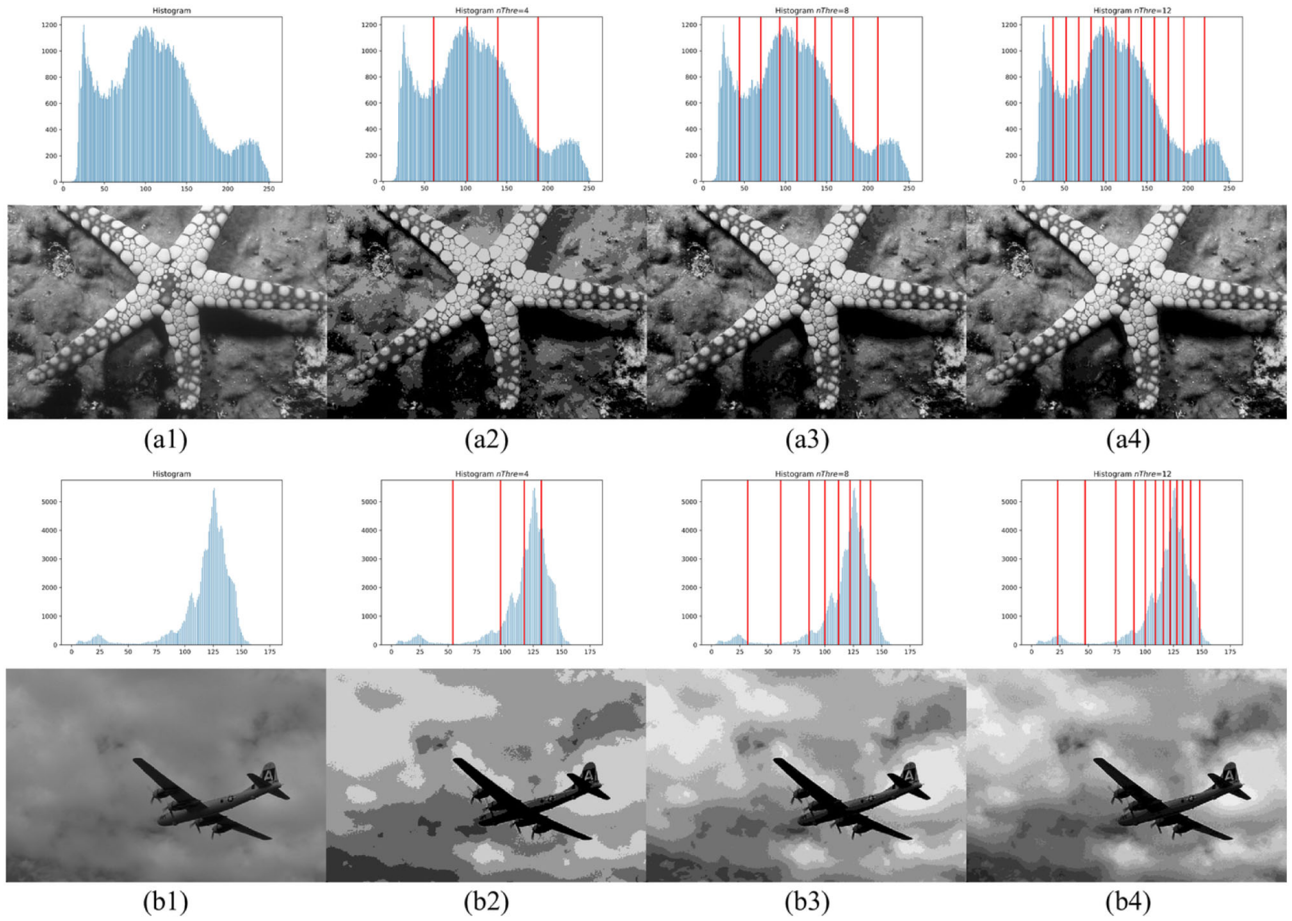


Fig. 4 The segmented images of image-1 and image-2 and corresponding histograms. Red lines represent thresholds (Color figure online)

The experimental results summarized in Tables 11 and 12 comprehensively demonstrate that the performance of the proposed SAO$_k$-AUS is competitive across 12 engineering problems.

Remarkably, SAO$_k$-AUS exhibits a remarkable and sustained superiority throughout the entirety of the numerical experiments, showcasing its prowess in outperforming competing algorithms across a majority of instances. In an extensive comparative analysis, SAO$_k$-AUS demonstrates a notable supremacy over its counterparts: SAP-DE in 11 instances, PPSO in 10 instances, MAEO in 7 instances, TSO in 9 instances, SHO in all 12 instances, SOA in 11 instances, COA in 7 instances, and even its predecessor SAO in 10 instances. This consistent dominance underscores the robustness and effectiveness of SAO$_k$-AUS in addressing the intricacies of engineering optimization problems.

To emulate real-world problem-solving scenarios comprehensively, the assessment extends beyond mere performance metrics to encompass the critical dimension of algorithmic stability. This crucial aspect is meticulously evaluated through metrics such as standard deviation (std), capturing the variability of solution quality, as well as the best and worst values attained by the algorithm. Through these performance indicators, a compelling observation emerges: SAO$_k$-AUS exhibits not only superior performance but also exceptional stability across a diverse array of engineering optimization problems. This demonstration of robustness and competitiveness establishes SAO$_k$-AUS as an excellent optimizer for real-world applications, reaffirming its position as a reliable tool for addressing complex optimization challenges in practical settings.

5.6 Performance analysis on the multilevel image segmentation task

In this study, we extend the application of SAO$_k$-AUS to tackle multilevel image segmentation tasks. By employing three critical metrics-fitness value, *PSNR*, and *SSIM*-we aim to provide a comprehensive evaluation of SAO$_k$-AUS's performance in this context. These metrics offer a holistic perspective on the quality of segmentation

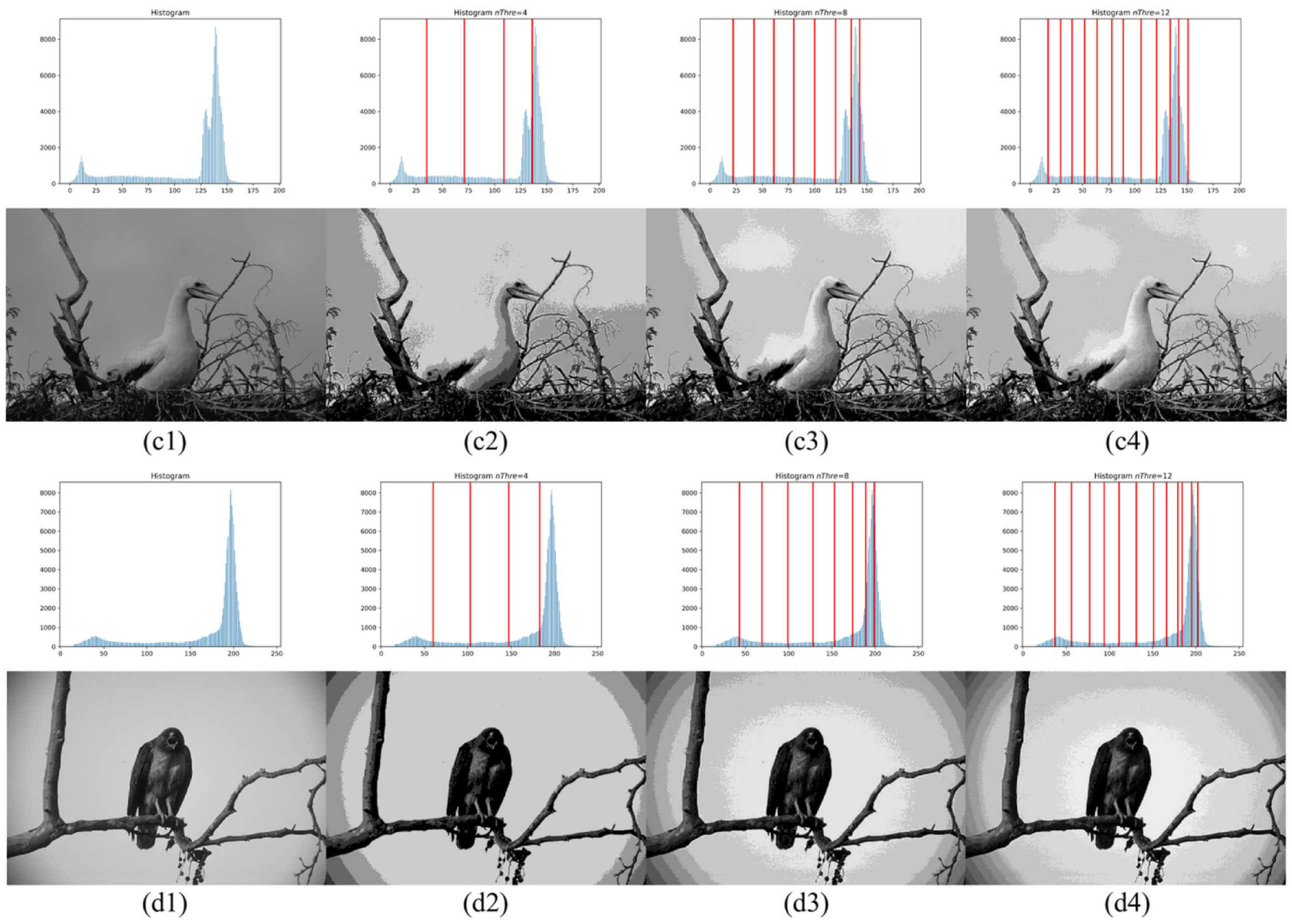


Fig. 5 The segmented images of image-3 and image-4 and corresponding histograms. Red lines represent thresholds (Color figure online)

achieved by the algorithm, covering aspects such as accuracy, fidelity, and SSIM with ground truth data, which are essential for assessing the effectiveness of segmentation algorithms in practical applications.

Our experimental results, summarized in Tables 13, 14, and 15, demonstrate the promising capabilities of SAO$_k$-AUS in multilevel image segmentation tasks. However, it's crucial to note that MAEO exhibits relatively better performance under specific conditions, particularly when the number of thresholds is set to four ($nThre = 4$). This observation prompts us to consider the broader context of algorithm performance, as stated by the No Free Lunch Theorem [48]. The No Free Lunch Theorem posits that any two stochastic optimization algorithms exhibit identical average performance across all conceivable problems. Therefore, when an algorithm demonstrates superiority in one category of problems, it is expected to exhibit a decline in performance in other categories to maintain an equilibrium of average performance. This theoretical framework offers valuable insights into understanding the relative performance variations observed in multilevel image segmentation tasks between SAO$_k$-AUS and MAEO.

In summary, while SAO$_k$-AUS showcases potential as a technique for addressing multilevel image segmentation tasks, the observed variations in performance underscore the importance of considering the inherent trade-offs and limitations of optimization algorithms across diverse problem domains. This extension of SAO$_k$-AUS highlights its versatility and adaptability to a wide range of real-world problems, while also emphasizing the need for nuanced

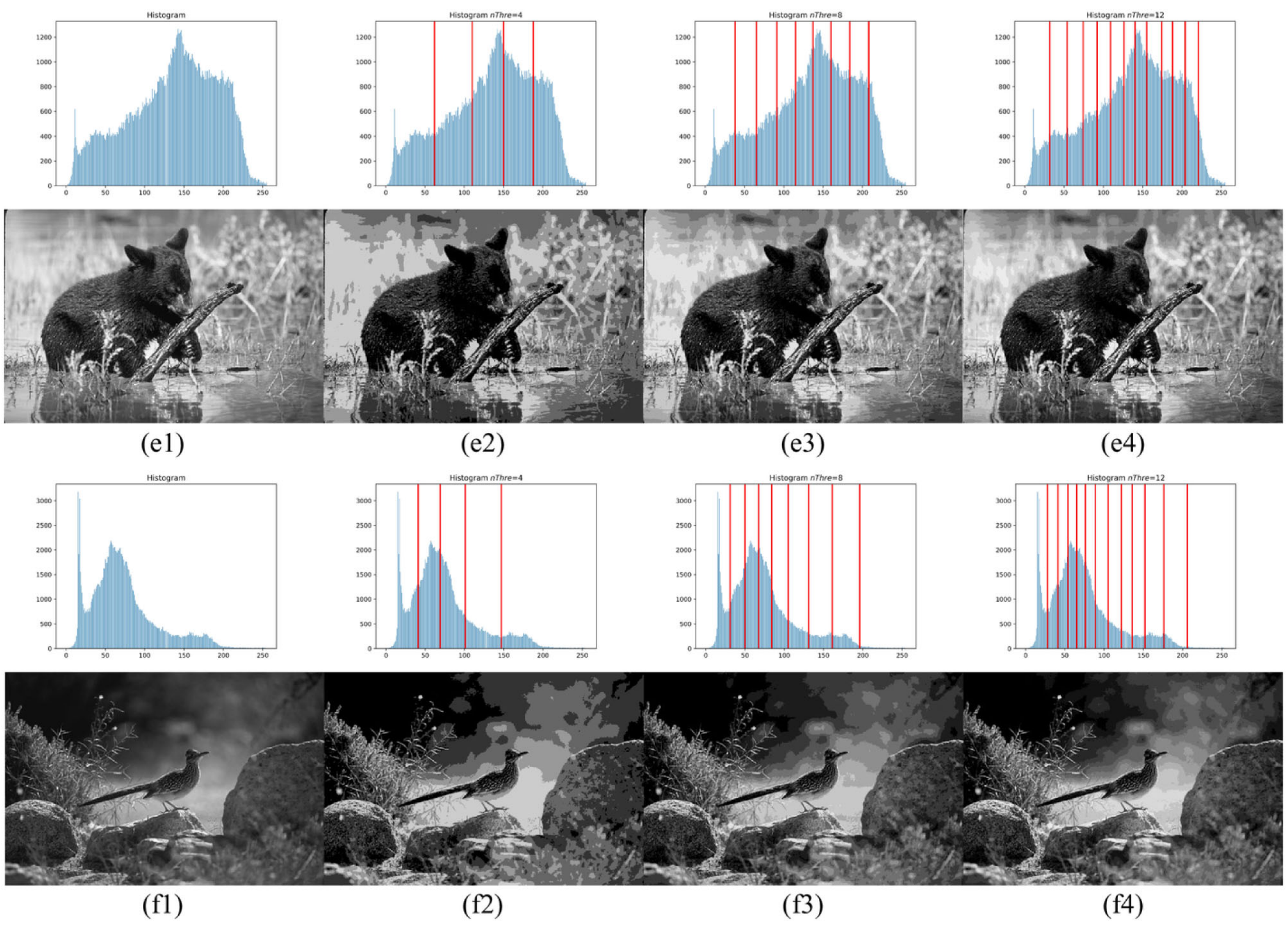


Fig. 6 The segmented images of image-5 and image-6 and corresponding histograms. Red lines represent thresholds (Color figure online)

evaluation and understanding of algorithmic performance in specific application scenarios.

6 Conclusion

In this paper, we propose an improved version of snow ablation optimization (SAO) named SAO$_k$-AUS. This improved version incorporates an AUS and the top-k survival mechanism. The AUS asynchronously integrates the current optimum update process after evaluation. When a superior optimum is discovered, this strategy promptly disseminates the knowledge to other individuals. The top-k survival mechanism ensures the quality of survived individuals. Comprehensive numerical experiments, encompassing CEC2013, and CEC2020 benchmark functions, 12 engineering problems, and multilevel threshold image segmentation tasks, prove practical evidence of the efficiency and effectiveness of our proposed SAO$_k$-AUS. The combination of these two strategies brings unexpected

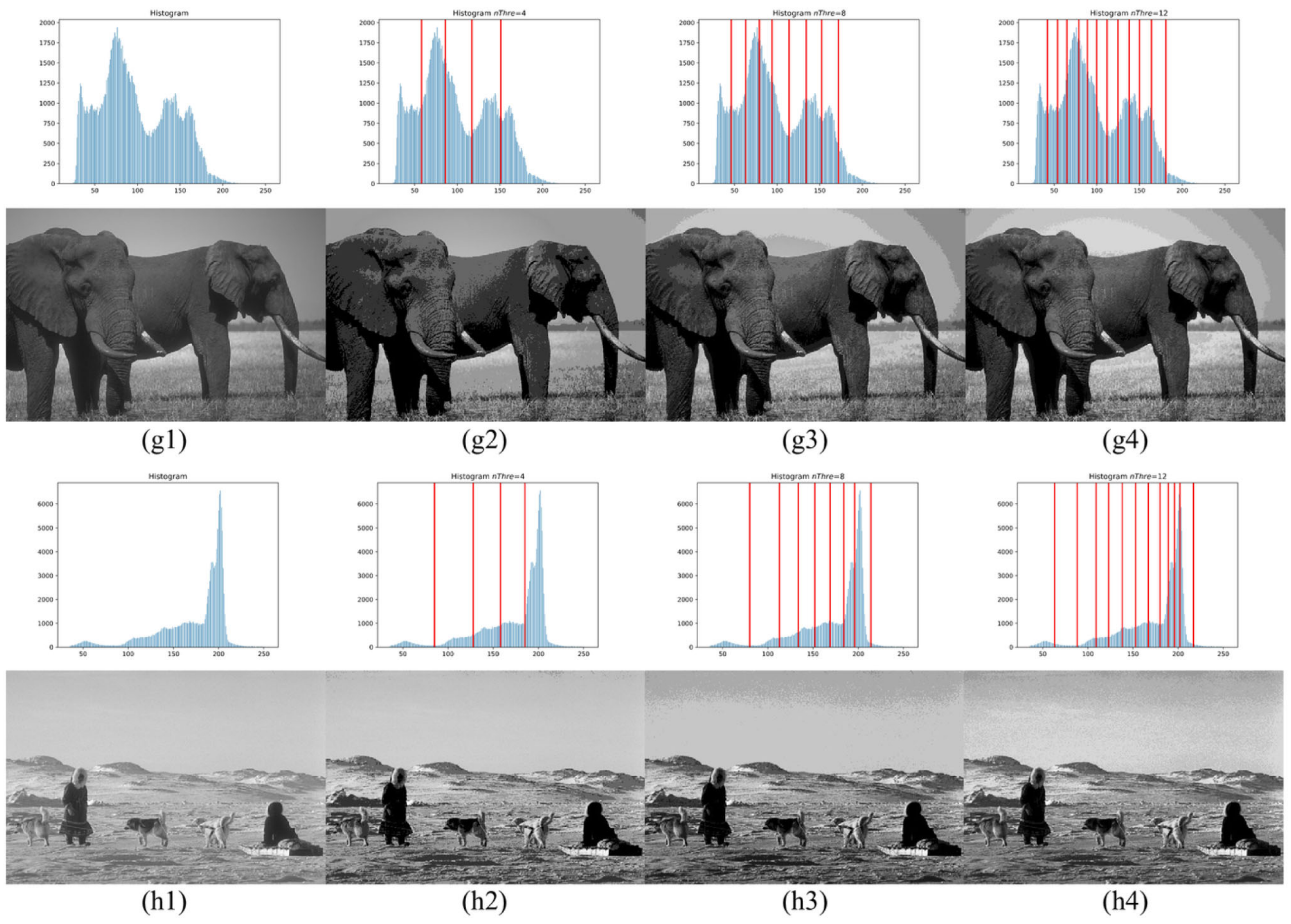


Fig. 7 The segmented images of image-7 and image-8 and corresponding histograms. Red lines represent thresholds (Color figure online)

improvements compared to employing each strategy in isolation. Therefore, we recommend employing both of these two proposed strategies simultaneously.

In the future, our focus will be on incorporating learning-based mechanisms into SAO and extending its applicability to various optimization tasks, including multi-objective optimization problems, binary optimization problems, expensive optimization problems, and large-scale optimization problems.

Author contributions Rui Zhong: Conceptualization, methodology, investigation, writing—original draft, writing—review and editing, and funding acquisition. Chao Zhang: Conceptualization and writing—review and editing. Jun Yu: Investigation, methodology, formal analysis, writing—review and editing, and project administration.

Funding This work was supported by JST SPRING Grant Number JPMJSP2119.

Data availability The source code of this research can be downloaded from https://github.com/RuiZhong961230/SAO_k-AUS.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Zhong, C., Li, G., Meng, Z., He, W.: Opposition-based learning equilibrium optimizer with levy flight and evolutionary population dynamics for high-dimensional global optimization problems. Expert Syst. Appl. **215**, 119303 (2023). https://doi.org/10.1016/j.eswa.2022.119303
2. Zhong, R., Yu, J., Chao, Z., Munetomo, M.: Surrogate ensemble-assisted hyper-heuristic algorithm for expensive optimization problems. Int. J. Comput. Intell. Syst. (2023). https://doi.org/10.1007/s44196-023-00346-y
3. Faridmehr, I., Nehdi, M.L., Davoudkhani, I.F., Poolad, A.: Mountaineering team-based optimization: a novel human-based metaheuristic algorithm. Mathematics (2023). https://doi.org/10.3390/math11051273
4. Singh, S., Singh, U.: A novel self-adaptive hybrid slime mould naked mole-rat algorithm for numerical optimization and energy-efficient wireless sensor network. Concurr. Comput. Pract. Exp. (2023). https://doi.org/10.1002/cpe.7809
5. Zhong, R., Peng, F., Yu, J., Munetomo, M.: Q-learning based vegetation evolution for numerical optimization and wireless

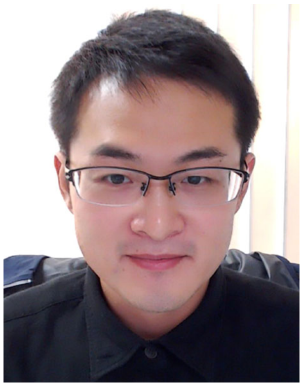
sensor network coverage optimization. Alex. Eng. J. **87**, 148–163 (2024). https://doi.org/10.1016/j.aej.2023.12.028

6. Abdel-Basset, M., El-Shahat, D., Jameel, M., Abouhawwash, M.: Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems. Artif. Intell. Rev. (2023). https://doi.org/10.1007/s10462-023-10403-9
7. Su, H., Zhao, D., Heidari, A.A., Liu, L., Zhang, X., Mafarja, M., Chen, H.: RIME: a physics-based optimization. Neurocomputing **532**, 183–214 (2023). https://doi.org/10.1016/j.neucom.2023.02.010
8. Zhong, R., Yu, J., Zhang, C., Munetomo, M.: SRIME: a strengthened rime with latin hypercube sampling and embedded distance-based selection for engineering optimization problems. Neural Comput. Appl. **36**, 6721–6740 (2024). https://doi.org/10.1007/s00521-024-09424-4
9. Juan, A., Keenan, P., Marti, R., McGarraghy, S., Panadero, J., Carroll, P., Oliva, D.: A review of the role of heuristics in stochastic optimisation: from metaheuristics to learn heuristics. Ann. Oper. Res. (2021). https://doi.org/10.1007/s10479-021-04142-9
10. Alorf, A.: A survey of recently developed metaheuristics and their comparative analysis. Eng. Appl. Artif. Intell. **117**, 105622 (2023). https://doi.org/10.1016/j.engappai.2022.105622
11. Khalid, O.W., Isa, N.A.M., Mat Sakim, H.A.: Emperor penguin optimizer: a comprehensive review based on state-of-the-art meta-heuristic algorithms. Alex. Eng. J. **63**, 487–526 (2023). https://doi.org/10.1016/j.aej.2022.08.013
12. Deng, L., Liu, S.: Snow ablation optimizer: a novel metaheuristic technique for numerical optimization and engineering design. Expert Syst. Appl. **225**, 120069 (2023). https://doi.org/10.1016/j.eswa.2023.120069
13. Martinec, J., Rango, A.: Parameter values for snowmelt runoff modelling. J. Hydrol. **84**(3), 197–219 (1986). https://doi.org/10.1016/0022-1694(86)90123-X
14. Zhong, R., Yu, J.: Dea2h2: differential evolution architecture based adaptive hyper-heuristic algorithm for continuous optimization. Clust. Comput. (2024). https://doi.org/10.1007/s10586-024-04587-0
15. Liu, Q., Li, N., Jia, H., Qi, Q., Abualigah, L.: Modified remora optimization algorithm for global optimization and multilevel thresholding image segmentation. Mathematics (2022). https://doi.org/10.3390/math10071014
16. Houssein, E., Hosny, M., Kamel, S., Hussain, K., Hashim, F.A.: Modified levy flight distribution algorithm for global optimization and parameters estimation of modified three-diode photovoltaic model. Appl. Intell. (2022). https://doi.org/10.1007/s10489-022-03977-4
17. Teo, J.: Exploring dynamic self-adaptive populations in differential evolution. Soft. Comput. **10**, 673–686 (2006). https://doi.org/10.1007/s00500-005-0537-1
18. Ghasemi, M., Akbari, E., Rahimnejad, A., Razavi, E., Ghavidel, S., Li, L.: Phasor particle swarm optimization: a simple and efficient variant of PSO. Soft. Comput. **23**, 9701–9718 (2019). https://doi.org/10.1007/s00500-018-3536-8
19. Menesy, A.S., Sultan, H.M., Korashy, A., Banakhr, F.A., Ashmawy, M.G., Kamel, S.: Effective parameter extraction of different polymer electrolyte membrane fuel cell stack models using a modified artificial ecosystem optimization algorithm. IEEE Access **8**, 31892–31909 (2020). https://doi.org/10.1109/ACCESS.2020.2973351
20. Lei, X., Tong, H., Huan, Z., Zhuo-Ran, Z., Bo, H., Andi, T.: Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization. Comput. Intell. Neurosci. (2021). https://doi.org/10.1155/2021/9210050
21. Zhao, S., Zhang, T., Ma, S., Wang, M.: Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. Appl. Intell. **53**, 11833–11860 (2022). https://doi.org/10.1007/s10489-022-03994-3
22. Dehghani, M., Trojovský, P.: Serval optimization algorithm: a new bio-inspired approach for solving optimization problems. Biomimetics (2022). https://doi.org/10.3390/biomimetics7040204
23. Dehghani, M., Montazeri, Z., Trojovská, E., Trojovský, P.: Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. Knowl. Based Syst. **259**, 110011 (2023). https://doi.org/10.1016/j.knosys.2022.110011
24. Abdel-Basset, M., Abdel-Fatah, L., Sangaiah, A.K.: Metaheuristic algorithms: a comprehensive review. In: Sangaiah, A.K., Sheng, M., Zhang, Z. (eds.) Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications. Intelligent Data-Centric Systems, pp. 185–231. Academic Press, New York (2018). https://doi.org/10.1016/B978-0-12-813314-9.00010-4
25. Bhandari, A.K., Rahul, K.: A novel local contrast fusion-based fuzzy model for color image multilevel thresholding using grasshopper optimization. Appl. Soft Comput. **81**, 105515 (2019). https://doi.org/10.1016/j.asoc.2019.105515
26. Bhandari, A.: A novel beta differential evolution algorithm-based fast multilevel thresholding for color image segmentation. Neural Comput. Appl. **32**, 4583–4613 (2020). https://doi.org/10.1007/s00521-018-3771-z
27. He, L., Huang, S.: An efficient krill herd algorithm for color image multilevel thresholding segmentation problem. Appl. Soft Comput. **89**, 106063 (2020). https://doi.org/10.1016/j.asoc.2020.106063
28. Lei, B., Fan, J.: Multilevel minimum cross entropy thresholding: a comparative study. Appl. Soft Comput. **96**, 106588 (2020). https://doi.org/10.1016/j.asoc.2020.106588
29. Lin, S., Jia, H., Abualigah, L., Altalhi, M.: Enhanced slime mould algorithm for multilevel thresholding image segmentation using entropy measures. Entropy (2021). https://doi.org/10.3390/e23121700
30. Köppen, M.: The curse of dimensionality. In: 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), vol. 1, pp. 4–8 (2000)
31. Zhong, R., Zhang, E., Munetomo, M.: Cooperative coevolutionary differential evolution with linkage measurement minimization for large-scale optimization problems in noisy environments. Complex Intell. Syst. (2023). https://doi.org/10.1007/s40747-022-00957-6
32. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979). https://doi.org/10.1109/TSMC.1979.4310076
33. Xiao, Y., Cui, H., Hussien, A.G., Hashim, F.A.: MSAO: a multistrategy boosted snow ablation optimizer for global optimization and real-world engineering applications. Adv. Eng. Inform. **61**, 102464 (2024). https://doi.org/10.1016/j.aei.2024.102464
34. Ismaeel, A.A.K., Houssein, E.H., Khafaga, D.S., Aldakhee, E.A., AbdElrazek, A.S., Said, M.: Performance of snow ablation optimization for solving optimum allocation of generator units. IEEE Access **12**, 17690–17707 (2024). https://doi.org/10.1109/ACCESS.2024.3357489
35. Jia, H., You, F., Wu, D., Rao, H., Wu, H., Abualigah, L.: Improved snow ablation optimizer with heat transfer and condensation strategy for global optimization problem. J. Comput. Des. Eng. **10**(6), 2177–2199 (2023). https://doi.org/10.1093/jcde/qwad096
36. Pandya, S.B., Kalita, K., Čep, R., Jangir, P., Chohan, J.S., Abualigah, L.: Multi-objective snow ablation optimization algorithm: an elementary vision for security-constrained optimal

power flow problem incorporating wind energy source with facts devices. Int. J. Comput. Intell. Syst. **17**(1), 1–30 (2024). https://doi.org/10.1007/s44196-024-00415-w
37. Lu, T., Li, H., Yang, Y., Yu, Z., Lei, Z., Gao, S.: Differential vectors empower snow ablation optimizer. In: 2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), vol. 11, pp. 1382–1386 (2023). https://doi.org/10.1109/ITAIC58329.2023.10409087
38. Liang, J., Qu, B., Suganthan, P., Hernández-Díaz, A.: Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou (2013)
39. Yue, C.T., Price, P.N.S.K.V.: Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore (2020)
40. Nguyen, T.: A framework of Optimization Functions using Numpy (OpFuNu) for optimization problems. Zenodo (2020). https://doi.org/10.5281/zenodo.3620960
41. Ezugwu, A., Agushaka, O., Abualigah, L., Mirjalili, S., Gandomi, A.: Prairie dog optimization algorithm. Neural Comput. Appl. **34**, 20017–20065 (2022). https://doi.org/10.1007/s00521-022-07530-9
42. Thieu, N.V.: ENOPPY: a Python library for engineering optimization problems. Zenodo (2023). https://doi.org/10.5281/zenodo.7953206
43. Van Thieu, N., Mirjalili, S.: MEALPY: an open-source library for latest meta-heuristic algorithms in python. J. Syst. Architect. **139**, 102871 (2023). https://doi.org/10.1016/j.sysarc.2023.102871
44. Zhong, R., Yu, J.: A novel evolutionary status guided hyper-heuristic algorithm for continuous optimization. Clust. Comput. (2024). https://doi.org/10.1007/s10586-024-04593-2
45. Coello Coello, C.A.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput. Methods Appl. Mech. Eng. **191**(11), 1245–1287 (2002). https://doi.org/10.1016/S0045-7825(01)00323-1
46. Sara, U., Akter, M., Uddin, M.S.: Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. J. Comput. Commun. **07**, 8–18 (2019). https://doi.org/10.4236/jcc.2019.73002
47. Yaguchi, K., Tamura, K., Yasuda, K., Ishigame, A.: Basic study of proximate optimality principle based combinatorial optimization method. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1753–1758 (2011). https://doi.org/10.1109/ICSMC.2011.6083925
48. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893

Rui Zhong received the B.Eng. degree from Huazhong Agricultural University, China in 2019, the M.Eng. degree from Kyushu University, Japan in 2022, and a Ph.D. at Hokkaido University, Japan in 2024. He is now a Specifically Appointed Assistant Professor at Information Initiative Center, Hokkaido University. His research interests include Evolutionary Computation, Large-scale Global Optimization, and Meta-/Hyper-heuristics.

Chao Zhang received his Ph.D. at Iwate University (Japan) in March 2017. He was a fulltime lecturer (a.k.a. assistant professor) at the Faculty of Engineering, University of Fukui (Japan) from 2017 to 2024. Starting from April 2024, he is currently a Specially Appointed Professor at the University of Toyama. His research interests include computer vision and machine learning. He is a member of the IEEE, IEICE, ITE, and IEEJ. More information can be found at https://www.labzhang.com/.


Jun Yu received a Bachelor degree from Northeastern University, China in 2014, and a Master degree and a doctorate from Kyushu University, Japan in 2017 and 2019, respectively. He is currently an Assistant Professor at Niigata University, Japan. His research interests include evolutionary computation, artificial neural networks, and machine learning.