SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2026, Vol. 69, Iss. 3, 132208:1–132208:17
https://doi.org/10.1007/s11432-025-4775-5

Adaptive small-family population-guided swarm intelligence optimization algorithm

Xintian WANG[1,2], Yongming HAN[1,2*], Chong CHU[3] & Zhiqiang GENG[1,2*]

[1] *College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China*
[2] *Engineering Research Center of Intelligent Process Systems Engineering, Ministry of Education, Beijing 100029, China*
[3] *Department of Biomedical Informatics, Harvard Medical School, Harvard University, Boston MA 02115, USA*

Received 17 February 2025/Revised 28 May 2025/Accepted 11 June 2025/Published online 2 February 2026

Abstract Ethylene, a cornerstone of the petrochemical industry, constitutes over 75% of petrochemical products and plays a vital role in the national economy. The ethylene cracking furnace, as the core production unit, directly determines the scale, output, and product quality of ethylene manufacturing. To optimize its operation, this paper proposes a novel adaptive small-family population-guided swarm intelligence optimization algorithm (ASPSIOA). The ASPSIOA operates in two distinct stages, including an early exploration stage and a late exploitation stage. Initially, the population is partitioned into small-family subgroups. During exploration, individuals are adaptively moved toward or away from their subgroup average position to identify promising regions in the search space. Subsequently, during exploitation, individuals are guided toward the global best solution, focusing the search on these identified promising areas. The performance of the ASPSIOA is rigorously evaluated on thirty benchmark functions and three semi-realistic engineering problems. Comparative results against six established algorithms demonstrate that the ASPSIOA achieves competitive performance with faster convergence speed and superior solution accuracy. Furthermore, the ASPSIOA is successfully applied to optimize an industrial ethylene cracking furnace by determining a periodic outlet temperature regulation strategy, resulting in a measurable increase in ethylene yield.

Keywords meta-heuristic algorithm, benchmark, adaptive small-family population-guided, swarm intelligence optimization algorithm, adaptive algorithm, process optimization, ethylene cracking furnace

Citation Wang X T, Han Y M, Chu C, et al. Adaptive small-family population-guided swarm intelligence optimization algorithm. Sci China Inf Sci, 2026, 69(3): 132208, https://doi.org/10.1007/s11432-025-4775-5

1 Introduction

Single-objective optimization problems form the foundation of numerous scientific and engineering challenges, consistently attracting significant research interest. To tackle these problems, a wide array of metaheuristic optimization algorithms has been developed and extensively studied. Over the past two decades, metaheuristic algorithms have gained widespread popularity due to their efficacy in solving complex and real-world problems [1]. Their appeal stems from key advantages such as flexibility, inherent parallelism, robustness, and relative ease of implementation. Furthermore, their derivative-free nature makes them particularly suitable for problems where gradient information is difficult or expensive to obtain, rendering them highly effective for black-box optimization scenarios [2], which versatility allows them to handle a broad spectrum of problems, including those with nonlinear, non-convex, discontinuous, or multimodal objective functions. Consequently, metaheuristic algorithms have become a vital tool in the optimization toolkit.

Metaheuristic optimization algorithms are divided into four main types, including the evolution-based algorithms, the swarm intelligence algorithms, the human-based algorithms and the physics/chemistry-based algorithms [3], which are widely used to solve real-world problems. Evolution-based algorithms are inspired by Darwin's theory of the natural selection. They mimic the process of evolution to improve a population over time and reach the best possible solution. Peng et al. used the genetic algorithm (GA) to improve sum rates [4], and Wang et al. used the differential evolution (DE) for the feature selection [5]. Swarm intelligence algorithms are based on the collective behavior of social animals like ants, bees, and other organisms that work together. These swarm intelligence algorithms mimic this behavior to find the best solutions, where each member of the group works together to accomplish a task. Numerous improvements and applications of swarm intelligence algorithms, such as the particle swarm optimization (PSO) [6] and the ant colony optimization [7], have been proposed. Zhang et al. enhanced

[*] Corresponding author (email: hanym@mail.buct.edu.cn, gengzhiqiang@mail.buct.edu.cn)

surgery assignment efficiency using the artificial bee colony algorithm [8], and Shen et al. applied the bat algorithm to identify parameters in grid-connected photovoltaic power generation models [9]. Jaisiva et al. integrated artificial neural networks with the firefly algorithm to address power system challenges [10], while Zhang et al. used the whale optimization algorithm to solve scheduling issues in workshops with automated vehicles [11]. Carbas et al. used the cuckoo optimization algorithm to minimize the design weight of complex steel structures [12].

Human-based algorithms are inspired by human behaviors, like teaching, social interactions, and problem-solving, which improve through trial and error, gradually finding better solutions. Yang et al. improved the teaching-learning algorithm to control UAV swarms for monitoring air quality and finding pollution sources [13]. Yu et al. used a similar algorithm to optimize photovoltaic system parameters [14]. Li et al. used the harmony search algorithm for image segmentation [15], and Majid applied the imperialist competitive algorithm to optimize antenna array spacing [16]. Liu et al. combined tabu search with evolutionary algorithms to avoid unnecessary searches [17], and also improved the interior search algorithm with reinforcement learning for energy-efficient job scheduling [18]. Zhang et al. used the firework algorithm to optimize storage assignments [19], while Sharifian et al. used a hybrid of the exchange market and grasshopper optimization algorithms for economic dispatch problems [20].

Physics/chemistry-based algorithms are inspired by natural physical laws and chemical reactions, such as simulated annealing (SA) [21], the gravitational search algorithm [22, 23], and the atomic orbital search algorithm [24], which are applied in the optimization of the residential roof top photovoltaic systems, and online product recommendation. However, in the above application of four meta-heuristic optimization algorithms, the problems of falling into the local optimal solution leading to poor solution results may occur. The evolution-based algorithms have the problem of difficult interpretation of results, the physics and chemistry-based algorithms have the problem of difficult to adjust parameters, the human-based algorithms have the problem of complex algorithms leading to high time cost, and the swarm intelligence optimization algorithms have the algorithmic effect of the unstable problem.

The no-free-lunch (NFL) theorem [25] formally establishes that no single algorithm can be universally superior across all optimization problems. This principle underscores the necessity for continuous algorithmic innovation and motivates the development of new metaheuristics tailored to specific problem characteristics.

Inspired by the NFL theorem and aiming to address the aforementioned limitations, this paper introduces a novel metaheuristic optimization algorithm of the adaptive small-family population-guided swarm intelligence optimization algorithm (ASPSIOA) for optimizing the operation of ethylene cracking furnaces. The ASPSIOA strategically divides the optimization process into two distinct stages including an early exploration stage and a late exploitation stage. The population is first partitioned into small-family groups. During exploration, individuals move adaptively toward or away from their group's average position to efficiently identify promising regions in the search space. During exploitation, individuals are guided toward the global best solution, leveraging inertial weights and adaptive exploration/exploitation factors to refine the search within those promising regions.

The performance of the ASPSIOA is rigorously evaluated on 30 benchmark functions and 3 semi-realistic engineering problems. Comparative studies against six established algorithms including the PSO, the GA, the SA, the Harmony Search (HS), the Artificial Bee Colony (ABC) and the Whale Optimization Algorithm (WOA) demonstrate that the ASPSIOA achieves competitive performance with faster convergence speed, higher solution accuracy, and improved stability. Ablation experiments are conducted to validate the effectiveness of the core small-family group strategy. Moreover, the ASPSIOA is successfully applied to a realistic industrial problem, which is optimizing an ethylene cracking furnace. By determining an optimal periodic outlet temperature regulation strategy, the ASPSIOA helps enhance ethylene yield, providing actionable insights for operational decision-making.

The key contributions of this work are summarized as follows.

(1) A new ASPSIOA is proposed, to employ a small-family population structure and a dual-stage search strategy for enhancing both global exploration and local exploitation capabilities.

(2) An adaptive weight adjustment mechanism is introduced to utilize a natural logarithm-based update rule for exploration and exploitation factors across different search stages, effectively balancing the search process.

(3) The practical utility of the ASPSIOA is demonstrated by optimizing the operation of an ethylene cracking furnace. The proposed algorithm identifies a fixed-cycle temperature adjustment strategy that significantly increases ethylene yield, showcasing its potential for improving industrial production efficiency.

The remainder of this paper is organized as follows. Section 2 details the proposed ASPSIOA methodology. Section 3 presents the experimental results and discussion on benchmark functions. Section 4 describes ablation studies. Section 5 provides a parameter sensitivity analysis. Sections 6 and 7 apply the ASPSIOA to semi-realistic and real-world industrial problems, respectively. Finally, Section 8 concludes the paper.

2 Methodology

2.1 Algorithmic framework

The ASPSIOA framework initiates by randomly generating an initial population. This population is then evenly divided into several small-family groups. And the optimization process unfolds in two sequential stages with an early exploration stage followed by a late exploitation stage, which allows the proposed algorithm to first broadly scout the solution space for promising regions before intensively searching within them.

2.2 Search strategy

2.2.1 *Exploration stage*

The primary objective of the early search stage is global exploration, aiming to identify regions with high potential within the solution space. During this stage, the position update for each individual is influenced primarily by the average position of its respective small-family group. An exploration factor, denoted as $w_1(iter)$, is employed to control the magnitude of this influence. The update mechanism allows individuals to move either closer to or farther from their group's centroid, thereby promoting diversity and extensive coverage. The mathematical models governing this stage are given by (1) and (2):

$$x_{ij}\left(iter+1\right) = x_{ij}\left(iter\right) \pm w_1(iter) \cdot \left(\bar{e}_{ij}\left(iter\right) - x_{ij}\left(iter\right)\right), \tag{1}$$

$$x_{ij}\left(iter+1\right) = x_{ij}\left(iter\right) \pm w_1(iter) \cdot \sin\left(\pi \cdot rand\right) \cdot \left(\bar{e}_{ij}\left(iter\right) - x_{ij}\left(iter\right)\right), \tag{2}$$

where x_{ij} denotes the j-th attribute of the i-th individual. $\bar{e}_{ij}$ denotes the best historical average value of the j-th attribute of the small-family group in which the i-th individual is a member. $w_1(iter)$ denotes the exploration factor. $iter$ denotes the current number of iterations.

The exploration factor $w_1(iter)$ decays adaptively with iterations to gradually shift focus from exploration to exploitation. Its value is calculated as follows:

$$w_1\left(iter\right) = \exp(-3.5 + 2 \times iter/max\_iter) \times 50, \tag{3}$$

where max_iter denotes the maximum number of iterations.

For each individual, four candidate offspring positions are generated using (1) and (2) (considering both positive and negative signs). The offspring with the best fitness value is selected as the individual's new position. The historical best position for each small-family group and the global best position are updated accordingly. This stage continues until a predefined fraction of the total iterations is completed (e.g., the first 1/5 of max_iter). The pseudocode for this stage is summarized in Algorithm 1.

Algorithm 1 Exploration stage of the ASPSIOA.

```
while ($iter \leqslant$ floor($1/5 \times max\_iter$)) do
    for individuals from each population do
        Calculate the average position of each small-family group, and use the historical optimal average position of the small-family group as the elite individual, denoted as $\bar{e}_{ij}(iter)$;
        $X(iter)_{offspring1,2}$ = Update individuals from each population by (1);
        $X(iter)_{offspring3,4}$ = Update individuals from each population by (2);
        Calculate the fitness values of the four offsprings and select the optimal one as the $X(iter+1)$;
        Calculate the fitness value for each population individual;
        Compare the size of the fitness values of the offspring individuals with the size of the fitness values of the current individuals and update them;
        Update the average individual position for each small-family group in the population and the optimal individual position for the entire population;
    end for
    $iter = iter + 1$.
end while
```

2.2.2 *Exploitation stage*

After promising regions are identified, the ASPSIOA enters the exploitation stage to conduct an intensive local search. In this stage, individual updates are guided by a combination of the individual's historical best position

(x_{ij}^{best}) and the global best position ($Best_j$). This guidance is modulated by an inertial weight w_2, an exploitation factor w_3, and a reintroduced exploration factor w_4.

The position update during exploitation follows one of the following three models:

$$\begin{aligned} x_{ij}\left(iter+1\right) =& w_2 \cdot x_{ij}^{best}\left(iter\right) + \frac{w_3}{10} \cdot \left(Best_j\left(iter\right) - x_{ij}\left(iter\right)\right) \\ & \pm w_4 \cdot \sin\left(\pi \cdot rand\right) \cdot \left(\bar{e}_{ij}\left(iter\right) - x_{ij}^{best}\left(iter\right)\right), \end{aligned} \tag{4}$$

$$\begin{aligned} x_{ij}\left(iter+1\right) =& w_2 \cdot x_{ij}^{best}\left(iter\right) + w_3 \cdot \cos\left(2\pi \cdot rand\right) \cdot \left(Best_j\left(iter\right) - x_{ij}\left(iter\right)\right) \\ & + w_4 \cdot \sin\left(\pi \cdot rand\right) \cdot \left(\bar{e}_{ij}\left(iter\right) - x_{ij}^{best}\left(iter\right)\right), \end{aligned} \tag{5}$$

$$\begin{aligned} x_{ij}\left(iter+1\right) =& w_2 \cdot x_{ij}^{best}\left(iter\right) + w_3 \cdot rand \cdot \left(Best_j\left(iter\right) - x_{ij}\left(iter\right)\right) \\ & + w_4 \cdot \sin\left(\pi \cdot rand\right) \cdot \left(\bar{e}_{ij}\left(iter\right) - x_{ij}^{best}\left(iter\right)\right), \end{aligned} \tag{6}$$

where x_{ij} denotes the j-th attribute of the i-th individual, $Best_j$ denotes the average value of the j-th attribute of the optimal individual, w_2 denotes the inertial weight, w_3 denotes the exploitation factor, and w_4 denotes the exploration factor.

According to (4)–(6), four offsprings of an individual can be obtained, and by comparing the fitness values of the four offsprings, the one with the smallest fitness value will be used as the final individual offspring of the exploitation behavior. The pseudo code is shown in Algorithm 2.

Algorithm 2 Exploitation stage of the ASPSIOA.

```
while floor(1/5 × $max\_iter$) < $iter$ < $max\_iter$ do
    for individuals from each population do
        Calculate the average position of each small-family group, and use the historical optimal average position of the small-family group as the elite individual, denoted as $\bar{e}_{ij}\left(iter\right)$;
        $X\left(iter\right)_{offspring1,2}$ = update individuals from each population by (4);
        $X\left(iter\right)_{offspring3}$ = update individuals from each population by (5);
        $X\left(iter\right)_{offspring4}$ = update individuals from each population by (6);
        Calculate the fitness values of the four offsprings and select the optimal one as the $X\left(iter+1\right)$;
        Calculate the fitness value for each population individual;
        Compare the size of the fitness values of the offspring individuals with the size of the fitness values of the current individuals and update them;
        Update the average individual position for each small-family group in the population and the optimal individual position for the entire population;
    end for
    $iter = iter + 1$.
end while
```

2.2.3 *Overall ASPSIOA process*

The complete ASPSIOA integrates the two stages sequentially. The proposed algorithm starts with the exploration stage as shown in Algorithm 1 for the initial portion of the iterations and then switches to the exploitation stage as shown in Algorithm 2 for the remainder. The overall procedure is summarized in Algorithm 3, and a corresponding framework diagram is depicted in Figure 1.

Algorithm 3 ASPSIOA.

```
Initialize the population $X_i$ $(i = 1, 2, \ldots, n)$;
Calculate the fitness value for each population individual;
Determine the average individual position for each small-family group and the global optimal position;
while $iter \leqslant$ floor(1/5 × $max\_iter$) do
    Execute pseudo-code in Algorithm 1;
end while
while floor(1/5 × $max\_iter$) < $iter$ and $iter$ < $max\_iter$ do
    Execute pseudo-code in Algorithm 2;
end while
return optimal individual position for the entire population.
```

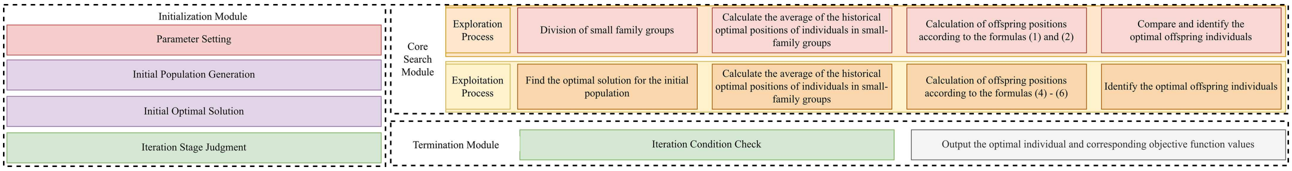
Initialization Module
Parameter Setting
Initial Population Generation
Initial Optimal Solution
Iteration Stage Judgment
Core Search Module
Exploration Process
Division of small family groups
Calculate the average of the historical optimal positions of individuals in small-family groups
Calculation of offspring positions according to the formulas (1) and (2)
Compare and identify the optimal offspring individuals
Exploitation Process
Find the optimal solution for the initial population
Calculate the average of the historical optimal positions of individuals in small-family groups
Calculation of offspring positions according to the formulas (4) - (6)
Identify the optimal offspring individuals
Termination Module
Iteration Condition Check
Output the optimal individual and corresponding objective function values

Figure 1 (Color online) ASPSIOA framework.

Table 1 The ASPSIOA parameter settings.

Parameter	Symbol	Value (no unit)
Population size	n	60
Maximum number of iterations	max_iter	500
Number of the individuals in the small family population	$npop$	6
Inertial weight in (4)–(6)	w_2	0.95
Exploration factor in (4)–(6)	w_3	2
Exploitation factor in (4)–(6)	w_4	2
Number of algorithms runs	$loop$	30

3 Results and discussion

In order to test the efficiency of the proposed ASPSIOA in this paper, the 30 standard test functions [26–31] are used for testing. The representative optimization algorithms from different kinds of metaheuristic optimization algorithms are selected for comparison, including the PSO [6], the GA [32], the SA [21], the HS [15], the ABC [8], and the WOA algorithms [33]. The performance of the algorithms is measured by comparing the best values they find for the test functions.

3.1 Parameterization

The ASPSIOA incorporates several key parameters that influence its search behavior, including the small-family group size ($npop$), the exploration factor (w_1), and the exploitation factor (w_3). To ensure statistical reliability and account for the stochastic nature of swarm intelligence algorithms, each benchmark function is independently solved 30 times by ASPSIOA and all compared algorithms. The mean and standard deviation (std) of the results from these 30 runs serve as the primary performance metrics.

After empirical testing and analysis of computational cost, the parameter set yielding the best overall performance is selected. The finalized parameter settings for ASPSIOA are listed in Table 1. For a fair comparison, the parameters of all competitor algorithms are set according to their original or widely adopted references.

3.2 Test functions

The proposed algorithm is tested on 30 standard benchmark functions, which are categorized into unimodal functions, multimodal functions, hybrid functions, and composition functions [31]. Unimodal functions, possessing a single global optimum, are utilized to assess an algorithm's convergence speed and local exploitation capability. Multimodal functions, with numerous local optima, test the algorithm ability to explore the search space and avoid premature convergence. Hybrid and composition functions present more complex landscapes, challenging the algorithm capacity to balance exploration and exploitation effectively [34]. A summary of the test functions, their variable dimensions, and value ranges is provided in Tables A1–A4 of Appendix A. Some of these functions are also shown in Figure 2 as two-dimensional plots.

3.3 Localized search capability test for the ASPSIOA

The results on unimodal functions (F1–F6) are presented in Table 2. The corresponding statistical summaries of the mean and standard deviation are visualized in Figures 3 and 4, respectively. The data indicate that ASPSIOA achieves the best mean values for most unimodal functions. Specifically, it attains the theoretical optimum for functions F1, F2, F3, and F5, and yields values very close to the optimum for F4 and F6. Furthermore, ASPSIOA consistently exhibits the lowest or nearly the lowest standard deviation across these tests. This performance demonstrates that ASPSIOA possesses strong and stable local exploitation capabilities, enabling it to converge rapidly and precisely towards the global optimum on unimodal benchmark functions with a single optimal region.

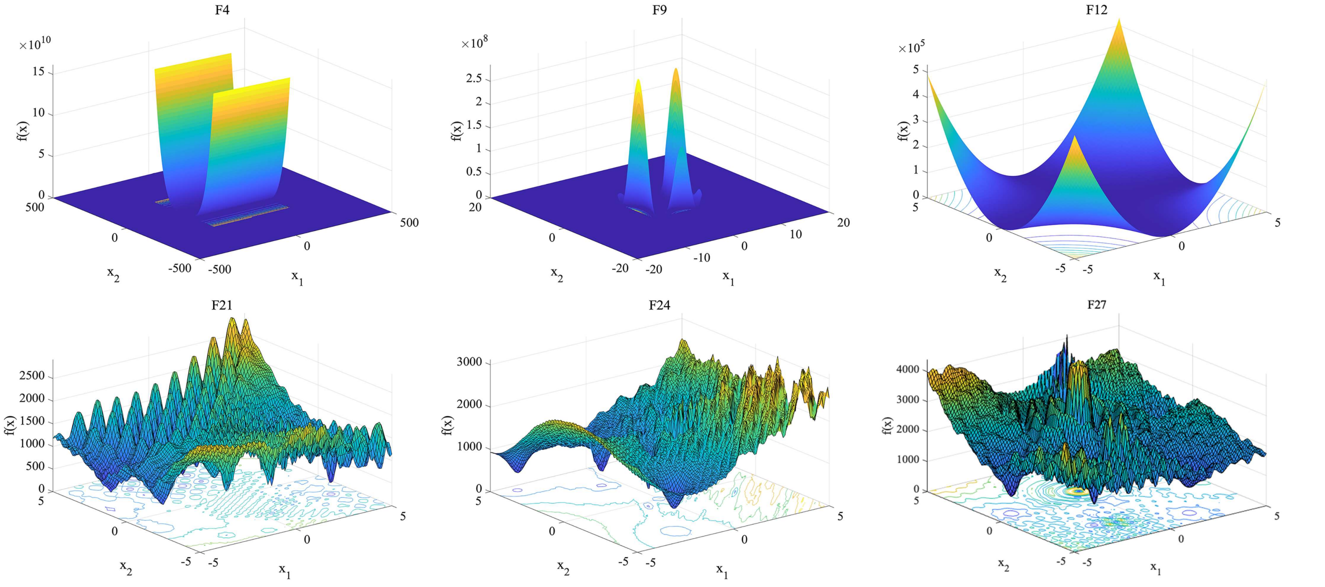

Figure 2 (Color online) Two-dimensional plots of some standard test functions.

Table 2 Comparison of optimization results for unimodal benchmark functions.

Function	Index	ASPSIOA	PSO	GA	SA	HS	ABC	WOA
F1	mean	0	2711.34	1131.71	–	54432.94	3695.72	0
	std	0	14850.62	0.08	–	3751.73	992.14	0
F2	mean	0	–	12.91	–	23942.44	31.1	0
	std	0	–	0.07	–	68936.95	3.00	0
F3	mean	0	2.83	52.14	87.10	73.40	7.17	59.08
	std	0	15.56	8.17	1.77	2.15	2.12	28.10
F4	mean	46.07	–	242097.01	–	–	704901.26	47.79
	std	0.18	–	412.54	–	–	318808.64	0.45
F5	mean	0	3278.97	1437.00	113681.30	53580.87	2231.63	0.37
	std	0	17959.64	0.37	7322.60	3533.47	486.73	0.18
F6	mean	0.01	12.17	0.82	313.47	178.38	0.45	0
	std	0	66.64	0.03	29.06	17.89	0.13	0

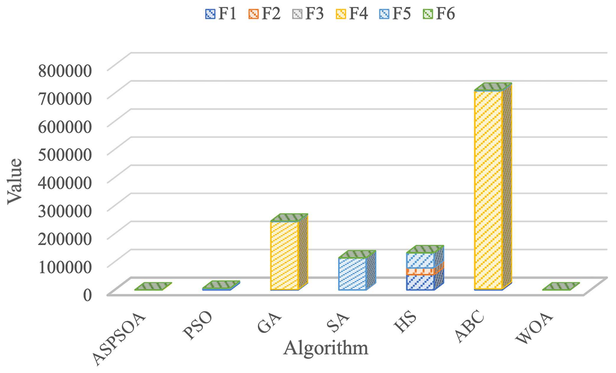

Figure 3 (Color online) Statistics of the mean values of the unimodal benchmark functions.

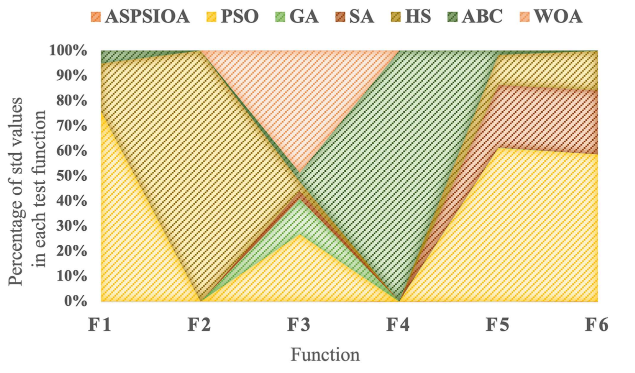

Figure 4 (Color online) Statistics of the std values of the unimodal benchmark functions.

3.4 Global search capability test for the ASPSIOA

The performance on multimodal functions (F7–F20) is summarized in Table 3. The statistical trends of the mean and standard deviation are illustrated in Figures 5 and 6. The ASPSIOA maintains its competitive edge in these

Table 3 Comparison of optimization results for multimodal and fixed-dimension multimodal benchmark functions.

Function	Index	ASPSIOA	PSO	GA	SA	HS	ABC	WOA
F7	mean	−5496.98	−18458.59	−168.92	−19518.81	−3823.93	−9439.25	−7910.82
	std	327.27	2793.06	925.24	2103.69	633.91	346.72	883.67
F8	mean	0	0	26.90	–	–	573.73	268.47
	std	0	0	147.34	–	–	18.39	18.71
F9	mean	0	0	0.67	7.66	20.50	19.33	10.07
	std	0	0	3.65	0.11	0.11	0.14	0.64
F10	mean	0	0.02	26.90	11.19	–	–	36.68
	std	0	0.06	147.35	0.16	–	–	7.75
F11	mean	0.99	1.46	0.03	1.00	1.49	1.00	1.00
	std	0	1.09	0.18	0.18	0.77	0	0
F12	mean	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0
F13	mean	−1.03	−1.03	−0.03	−1.03	−1.02	−1.03	−1.03
	std	0	0	0.19	0.19	0.01	0	0
F14	mean	0.39	0.40	0.01	0.40	0.40	0.40	0.40
	std	0	0	0.07	0.07	0	0	0.03
F15	mean	3.00	3.00	0.10	3.00	3.18	3.02	3.00
	std	0	0	0.55	0.55	0.18	0.01	0.22
F16	mean	−3.86	−3.85	−0.13	−3.86	−3.85	−3.86	−3.86
	std	0	0	0.70	0.71	0.01	0	0.24
F17	mean	−3.32	−3.25	−0.09	−3.32	−3.00	−3.04	−3.29
	std	0	0.09	0.48	0.61	0.10	0.05	0.05
F18	mean	−7.07	−8.71	−0.16	−2.62	−3.04	−6.13	−9.90
	std	0.35	2.46	0.85	1.85	1.49	2.701	1.37
F19	mean	−7.73	−8.92	−0.18	−2.76	−2.87	−8.98	−10.18
	std	1.34	2.70	0.97	1.90	0.74	0.59	1.22
F20	mean	−9.92	−8.77	−0.06	−10.35	−2.83	−8.99	−10.28
	std	1.69	3.08	0.34	1.92	1.03	0.72	1.40

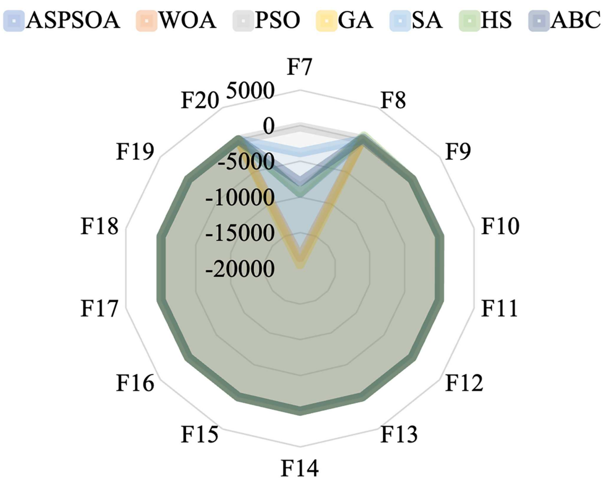

Figure 5 (Color online) Statistics of the mean values of the multimodal and fixed-dimension multimodal benchmark functions.

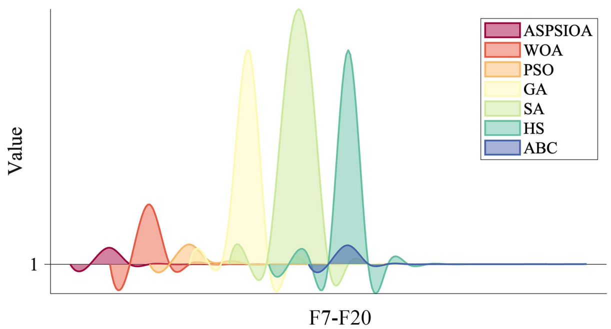

Figure 6 (Color online) Statistics of the std values of the multimodal and fixed-dimension multimodal benchmark functions.

more challenging scenarios, which secures the best mean results for the majority of the multimodal and fixed-dimension multimodal functions. More importantly, its standard deviation remains among the lowest, indicating robust performance across different runs. These results confirm that the small-family group mechanism and the dual-stage search strategy effectively preserve population diversity, which allows the ASPSIOA to thoroughly explore complex search spaces, evade local optima, and reliably locate promising regions, thereby exhibiting excellent global exploration capability.

Table 4 Comparison of optimization results for the combined benchmark functions.

Function	Index	ASPSIOA	PSO	GA	SA	HS	ABC	WOA
F21	mean	651.48	972.89	1340.15	1083.81	1356.19	1280.07	1039.09
	std	62.68	261.85	65.89	87.08	54.91	41.71	157.43
F22	mean	546.46	761.51	1252.33	917.38	1228.69	1076.06	836.00
	std	41.53	118.41	130.45	28.59	79.85	77.77	98.81
F23	mean	550.83	765.63	1446.60	995.50	1315.47	1208.30	820.44
	std	29.95	78.35	118.78	32.46	85.50	73.62	95.56
F24	mean	910.00	1100.12	1385.16	1230.31	1440.72	1376.88	1049.47
	std	0	186.01	109.44	41.61	26.26	25.65	32.49
F25	mean	910.00	1104.52	1375.36	1208.56	1433.97	1368.56	1052.07
	std	0	201.96	109.66	60.89	31.43	34.92	25.88
F26	mean	910.00	1110.36	1347.64	1208.25	1436.81	1379.04	1048.34
	std	0	193.67	104.51	58.15	37.26	29.63	22.47
F27	mean	1593.89	1742.24	1761.15	1667.63	1920.13	1849.52	1761.84
	std	81.52	36.64	113.52	1.30	30.99	27.85	26.90
F28	mean	1459.21	1574.68	1903.90	1787.21	2048.20	1980.14	1684.50
	std	42.83	80.78	155.56	20.48	58.13	62.55	88.15
F29	mean	1564.41	1657.59	1704.86	1700.93	1892.68	1828.60	1664.66
	std	21.35	30.52	117.26	7.84	39.95	23.09	24.75
F30	mean	1556.18	1646.73	1731.47	1703.77	1898.50	1829.74	1660.33
	std	16.09	31.93	128.75	24.88	38.27	27.53	28.59

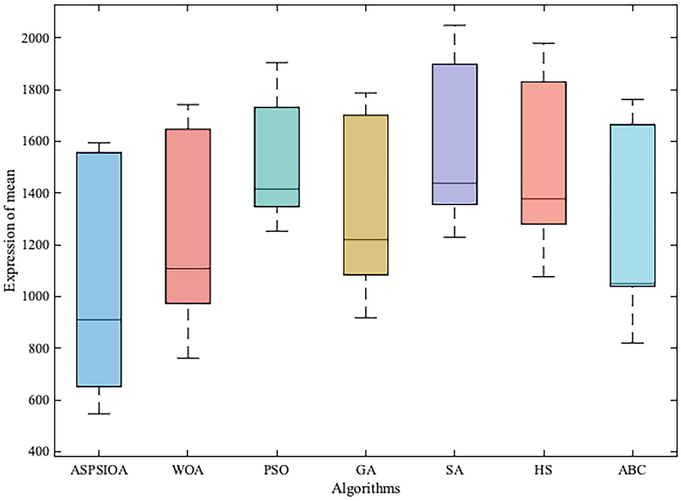

Figure 7 (Color online) Statistics of the mean values of the composite benchmark functions.

3.5 Overall competence test of the ASPSIOA

The results for the more complex hybrid and composition functions (F21–F30) are listed in Table 4. The statistical overview is provided in Figures 7 and 8. The ASPSIOA continues to deliver superior performance on these composite functions, obtaining the best mean values for most test cases. This success can be attributed to its inherent design for balancing exploration and exploitation.

$$x_{ij}\left(iter+1\right) = x_{ij}\left(iter\right) \pm w_1 \cdot \left(\bar{x}_{ij}\left(iter\right) - x_{ij}\left(iter\right)\right) + 0 * \left(Best_j\left(iter\right) - x_{ij}\left(iter\right)\right), \tag{7}$$

$$x_{ij}\left(iter+1\right) = x_{ij}\left(iter\right) \pm \sin\left(\pi \cdot rand\right) \cdot w_1 \cdot \left(\bar{x}_{ij}\left(iter\right) - x_{ij}\left(iter\right)\right) + 0 * \left(Best_j\left(iter\right) - x_{ij}\left(iter\right)\right). \tag{8}$$

According to (7) and (8), it can be assumed that the exploration factor is w_2 and the exploitation factor is 0. The values of the exploration factor and the exploitation factor of the ASPSIOA in different stages ensure the ability of the optimization algorithm to balance the exploration and exploitation stages.

To statistically validate the superiority of the ASPSIOA, the Wilcoxon signed-rank test is employed for pairwise comparison against each competitor algorithm across all benchmark functions. The resulting p-values are reported

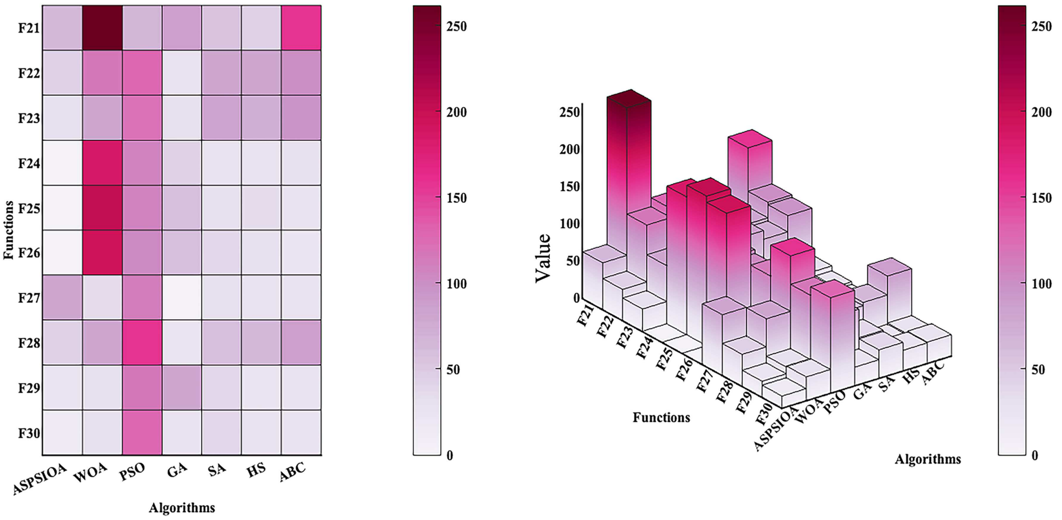

Figure 8 (Color online) Statistics of the std values of the composite benchmark functions.

Table 5 Results of the Wilcoxon signed-rank test. Note: w/t/l denote the number of cases where ASPSIOA performs better/worse/the same as the comparison algorithms in the Wilcoxon signed-rank test, and the numerical results where ASPSIOA outperforms the comparison algorithms are shown in italic and bold in the table.

Index	WOA	PSO	GA	SA	HS	ABC	w/t/l
mean	***9.40E−03***	***6.34E−06***	***3.88E−04***	***1.73E−06***	***2.83E−04***	***2.41E−02***	6/0/0
std	***8.14E−05***	***9.63E−04***	7.81E−02	***7.27E−03***	***3.50E−02***	***3.38E−02***	5/1/0

in Table 5. A p-value less than or equal to 0.05 provides strong evidence that the performance difference between the two algorithms is statistically significant. As shown in Table 5, the ASPSIOA significantly outperforms all six competitor algorithms in terms of the solution quality (mean). The results for the stability (std) are also highly favorable. This statistical test conclusively establishes the ASPSIOA as a highly competitive optimizer.

3.6 Analysis of convergence behavior

The convergence profiles of the ASPSIOA and the compared algorithms on selected functions are plotted in Figures 9 and 10. The convergence curves reveal two key characteristics of the ASPSIOA. First, it demonstrates a rapid initial convergence rate, evidenced by sharp declines in fitness value during the early iterations (e.g., in F1, F5, F11). This aligns with its effective exploration phase which quickly identifies high-potential regions. Second, in the later stages, the ASPSIOA exhibits refined local search behavior, gradually approaching the optimal value with high precision as shown in Figure 10. This balanced convergence behavior, combining swift initial progress with accurate final refinement, underscores the effectiveness of its staged search strategy.

3.7 Discussion

The experimental results collectively affirm that the ASPSIOA exhibits strong capabilities in exploration, exploitation, and convergence. The ASPSIOA's robust exploration stems from the small-family group strategy. Partitioning the population and using subgroup centroids as movement references encourages diverse sampling of the search space, especially during the early stage. The adaptive decay of the exploration factor $w_1(iter)$ further modulates this process.

Its high exploitation ability is derived from the focused search in the later stage. Guiding individuals based on their historical best position (x_{ij}^{best}) and the global best ($Best_j$), modulated by w_2, w_3, and w_4, enables precise refinement within promising regions. The balanced convergence is achieved by the explicit separation and adaptive transition between the two stages. While exploitation is emphasized later, the retention of group-level information prevents excessive greediness.

However, certain limitations are observed and align with the NFL theorem. For instance, on the high-dimensional asymmetric function F6, the ASPSIOA performance is slightly inferior to the WOA algorithm as shown in Table 2, which may be attributed to limited information exchange between isolated small-family groups when facing highly irregular landscapes. Similarly, the fixed subpopulation size might limit coverage in problems with an extremely

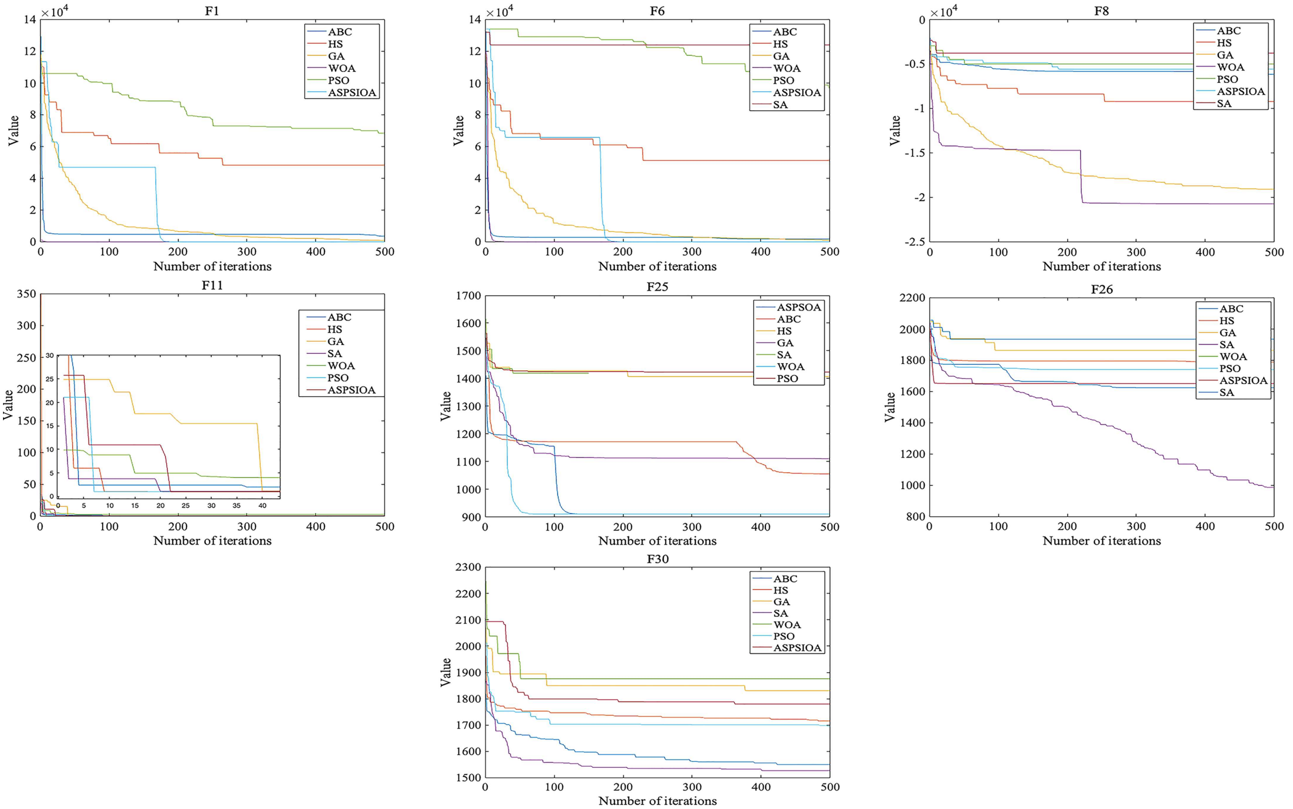

Figure 9 (Color online) The convergence plots of the compared optimization algorithms on some test functions.

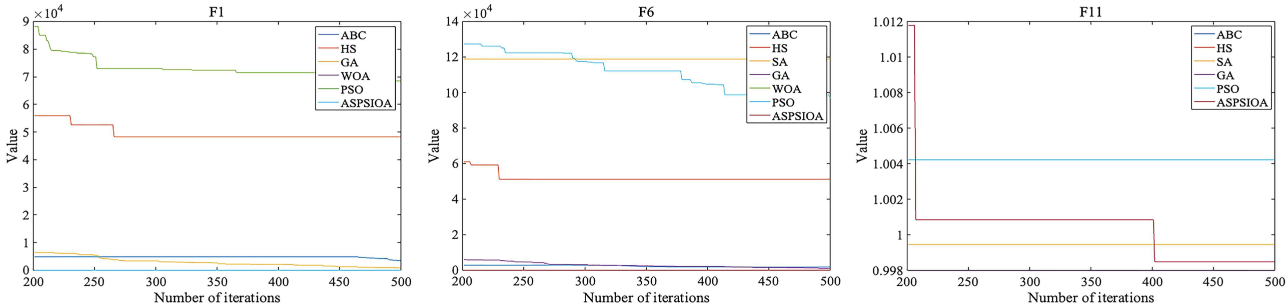

Figure 10 (Color online) The convergence plots of the compared optimization algorithms on some test functions in the later stages of the search process.

large number of local optima. Future work could investigate adaptive mechanisms for subpopulation size or enhanced inter-group communication strategies, albeit with careful consideration of added computational cost.

4 Ablation experiments and results

To deeply analyze the contribution of the core small-family group strategy proposed in this paper, a set of ablation experiments is conducted. A variant of the algorithm, termed the adaptive swarm intelligence optimization algorithm (ASIOA), is constructed by removing the small-family group mechanism, where the historical optimal position of the entire population is used as the elite guidance ($\epsilon_{ij}(iter)$), while the structure of the individual position update formulas remains identical to that of the ASPSIOA.

The performance of the ASPSIOA and the ASIOA is compared on the same suite of benchmark functions, with each algorithm executed for 30 independent runs. The complete numerical results are recorded in Table B1 of Appendix B. For brevity, test functions where the results between the two algorithms are identical (and thus non-comparative) are omitted from the summary table.

The results consistently demonstrate the superiority of the ASPSIOA over the ASIOA across most functions. The small-family group strategy, by implementing a form of elite reverse learning at the subgroup level, effectively expands the search range and generates more diverse candidate solutions. This is particularly beneficial for solving nonconvex problems with multiple extreme points. Compared to the ASIOA, the ASPSIOA maintains superior population diversity throughout the search process, which aids in escaping local optima, without incurring excessive computational overhead. This ablation study confirms that the small-family group structure is a crucial and effective component of the proposed algorithm.

5 Parameter sensitivity analysis

The ASPSIOA involves several critical parameters whose settings can significantly impact its global search capability, local refinement accuracy, and ability to escape local optima. This section systematically analyzes the influence of these parameters.

5.1 Number of individuals in small-family groups (*npop*)

The parameter *npop* determines the granularity of population partitioning. A smaller *npop* value increases the number of groups, potentially enhancing global exploration breadth but may reduce the guiding influence of each group's centroid on local search. Conversely, a larger *npop* strengthens local guidance but may compromise diversity. To balance this trade-off, *npop* is tested at values of 4, 6, 10, 12, and 15.

The experimental results for different *npop* settings are detailed in Table B2 of Appendix B. Functions where results are identical across settings are omitted. Analysis of Table B2 indicates that setting $npop = 6$ provides the best compromise, yielding high optimization accuracy across various function types while maintaining robust performance stability. Therefore, $npop = 6$ is adopted as the default value.

5.2 Exploration factor and exploitation factor

Factors affect the global and local search capabilities of the ASPSIOA. The exploration factor mainly affects the search range of the ASPSIOA, and the exploitation factor mainly affects the convergence of the ASPSIOA. Therefore, it is crucial to adjust the value of the factor and balance the exploration and exploitation capabilities of the ASPSIOA. This section discusses the impact of factor sizes at different stages on the algorithm, setting different sizes and combining them.

5.2.1 *Exploration factor $w_1(iter)$ in (2)*

According to Section 2, $w_1(iter)$ varies with the number of iterations. Combining (1) and (2) for analysis, $w_1(iter)$ is too high for the proposed algorithm to perform global search, and $w_1(iter)$ is too low for the proposed algorithm to converge. This section discusses the influence of the coefficients of the exploration factor on the effect of the proposed algorithm, morphing (3) into

$$w_1\,(iter) = \exp(b_1 + b_2 \times iter/max\_iter) \times b_3. \tag{9}$$

To balance exploration and exploitation, b_1 is tested at values of -2.5, -3.5, and -4.5, b_2 is tested at values of 1, 2, and 3, and b_3 is tested at values of 40, 50, and 60. From the permutation principle, it can be seen that there are 27 sets of coefficient settings for $w_1(iter)$, as shown in Figure 11. By observing Figure 11, it can be found that $w_1(iter)$ has a more reasonable value size throughout the exploration stage when b_1 takes the value of -3.5. Combining different coefficient sizes, the experimental results are shown in Table B3 of Appendix B, where the test functions that are not comparable (i.e., the results are the same) are deleted.

From Table B3, it can be found that when (b_1, b_2, b_3) is taken to be (-3.5, 2, 50), the overall optimization accuracy of the ASPSIOA is the highest, while the robustness of the ASPSIOA is also the most stable.

5.2.2 *Exploration factor w_3 in (4)–(6)*

In the exploitation stage, the exploration factor balances the global search capability of the ASPSIOA. In this section, the w_3 is tested at values of 0.5, 0.8, 2, 3, and 3.5, and the experimental results are shown in Table B4 of Appendix B, where the test functions that are not comparable (i.e., the results are the same) are deleted.

From Table B4, it can be found that when w_3 is taken to be 2, the overall optimization accuracy of the ASPSIOA is the highest, while the robustness of the ASPSIOA is also the most stable.

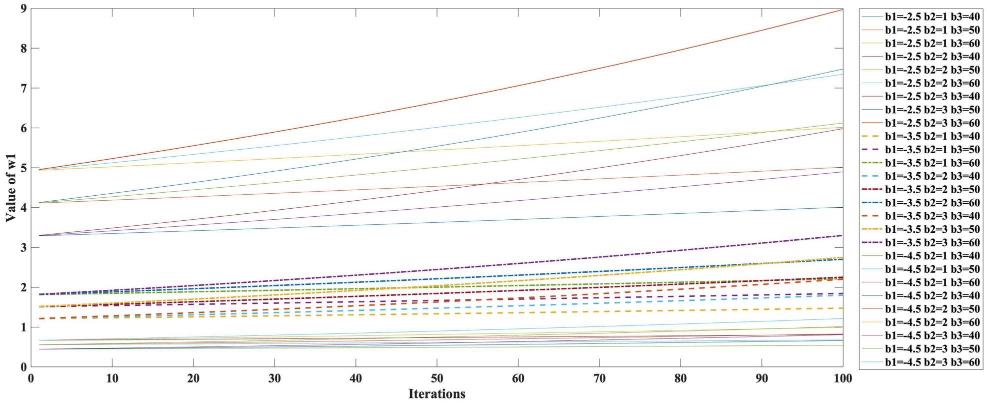

Figure 11 (Color online) Different sets of coefficient settings for $w_1(iter)$.

5.2.3 *Exploitation factor w_4 in (4)–(6)*

In the exploitation stage, the exploitation factor improves the local search capability of the ASPSIOA. In this section, w_4 is tested at values of 0.5, 0.8, 2, 3, and 3.5, and the experimental results are shown in Table B5 of Appendix B, where the test functions that are not comparable (i.e., the results are the same) are deleted.

From Table B5, it can be found that when w_4 is taken to be 2, the overall optimization accuracy of the ASPSIOA is the highest, while the robustness of the ASPSIOA is also the most stable.

5.3 Inertial weight w_2 in (4)–(6)

In the exploitation stage, the inertial weight improves the global search capability of the ASPSIOA. In this section, w_2 is tested at values of 0.5, 0.7, 1.2, 2, and 3, and the experimental results are shown in Table B6 of Appendix B, where the test functions that are not comparable (i.e., the results are the same) are deleted.

From Table B6, it can be found that the overall optimization accuracy of the ASPSIOA may be the highest and the robustness of the ASPSIOA may be the most stable when w_2 takes the value of [0.5, 1.5]. In order to reduce the possibility of the ASPSIOA falling into a local optimal solution, this paper sets the value of w_2 to 0.95. Experimental results show that when w_2 is 0.95, the overall effect of the ASPSIOA is better than when w_2 is 0.7. Therefore, w_2 is taken as 0.95.

6 Application of the ASPSIOA to semi-realistic problems

To further validate the robustness and practical applicability of the ASPSIOA, it is tested on three constrained engineering optimization problems selected from the CEC2020 benchmark [35]. These problems involve mixed variable types (continuous, integer, discrete) and nonlinear constraints, representing common challenges in mechanical design. A constraint-handling method is required due to the presence of limitations. Among various penalty function methods (static, dynamic, annealing, adaptive, death penalty) [36], the death penalty method is adopted for its simplicity in this study. The death penalty method assigns an extremely poor objective function value (for minimization) to any solution that violates a constraint, effectively eliminating it from the feasible candidate pool.

6.1 Planetary gear train design optimization problem

This case involves the design of a compound planetary gear train, where the objective is to minimize the maximum error between the achieved and target gear ratios (or speed ratios) [37]. The optimization requires calculating the number of teeth for six gears, subject to 11 constraints encompassing geometric compatibility, assembly conditions, and adjacency requirements. The detailed mathematical formulation of this problem, including the objective function and all constraints, is provided in Appendix C.1.

6.2 Rolling element bearing

This problem focuses on maximizing the dynamic load-carrying capacity of a rolling element bearing, a critical factor determining its service life [38]. While the external geometry is simple, the internal geometry significantly influences performance metrics like stress distribution and fatigue life. The optimization involves five primary design variables

Table 6 Comparison of optimization results for semi-realistic problems.

Function	Fmin	Index	ASPSIOA	WOA	PSO	GA	SA	HS	ABC
F31	5.26E−01	mean	5.30E−01	6.45E−01	–	1.01E+00	1.62E+00	–	5.55E−01
		std	5.20E−03	1.02E−01	–	3.16E−02	6.22E−01	–	4.95E−02
F32	1.46E+04	mean	1.03E+04	4.83E+03	1.50E+04	1.60E+04	2.09E+04	–	4.83E+03
		std	3.70E−12	2.78E−12	1.55E+04	6.60E−12	3.45E+03	–	2.78E−12
F33	2.64E+00	mean	2.64E+00	2.64E+00	2.70E+00	2.74E+00	3.97E+00	4.55E+00	2.89E+00
		std	2.29E−15	0.00E+00	6.20E−02	1.18E−15	2.71E−01	2.71E−01	2.74E−01

and five additional internal geometry parameters (e.g., e, ϵ, K_{Dmax}, K_{Dmin}), which are treated as variables during optimization. The design is subject to nine nonlinear constraints related to manufacturing and kinematic feasibility. A complete description of this problem is available in Appendix C.2.

6.3 Topology optimization

The topology optimization problem aims to find the optimal material distribution within a predefined design domain, given a set of loads, constraints, and performance objectives [39]. The goal is to minimize compliance while satisfying a volume fraction constraint. The detailed formulation utilizing the solid isotropic material with the penalization method is presented in Appendix C.3.

6.4 Results on semi-realistic problems

The ASPSIOA is applied to solve the three problems described above. Each algorithm is run independently 30 times. The parameter settings for all algorithms remain consistent with those in Section 3. The statistical results are summarized in Table 6. The results in Table 6 demonstrate that ASPSIOA achieves highly competitive outcomes on these semi-realistic, constrained problems. It obtains the best mean values for all three cases. The consistently low standard deviation further confirms the stability and robustness of ASPSIOA when handling complex, constrained search spaces, underscoring its potential for real-world engineering design applications.

7 Application of the ASPSIOA to the realistic problem

Ethylene is very important for the petrochemical industry and the economy, as it helps measure a country's industrial progress [40, 41]. With economic growth and a rising population, the global demand for ethylene keeps increasing [42, 43], making it one of the most in-demand chemicals in the world [44]. The ethylene cracking furnace is a key part of the production process, and the flow chart of the ethylene cracking process is shown in Figure 12, which is a complex system that needs to be adjusted based on different production needs.

7.1 Ethylene cracking furnace model

The industrial case study employs a model of an SL-I type ethylene cracking furnace using naphtha as feedstock. The furnace operates in cycles, typically lasting around 63 days. A major operational challenge is coking, where carbonaceous by-products gradually deposit on the inner walls of the radiant tubes. This coke layer acts as an insulator, reducing heat transfer efficiency and causing a gradual drop in the internal reaction temperature, which in turn decreases ethylene yield over time.

To counteract this decline, a common practice is to periodically increase the furnace outlet temperature (COT) to compensate for the insulating effect. This study formulates an optimization problem to determine an optimal fixed-cycle temperature regulation strategy. Specifically, the goal is to find the best magnitude of temperature adjustment to apply at regular 7-day intervals, maximizing the average ethylene yield over the entire operating cycle while respecting all operational constraints.

The optimization objective is to maximize the average daily ethylene yield, formulated as follows:

$$Z = \max\left(\frac{1}{D}\sum_{i=1}^{D} x_i\right), \tag{10}$$

where D is the total number of cracking days in a cycle, and x_i is the ethylene yield on the i-th day.

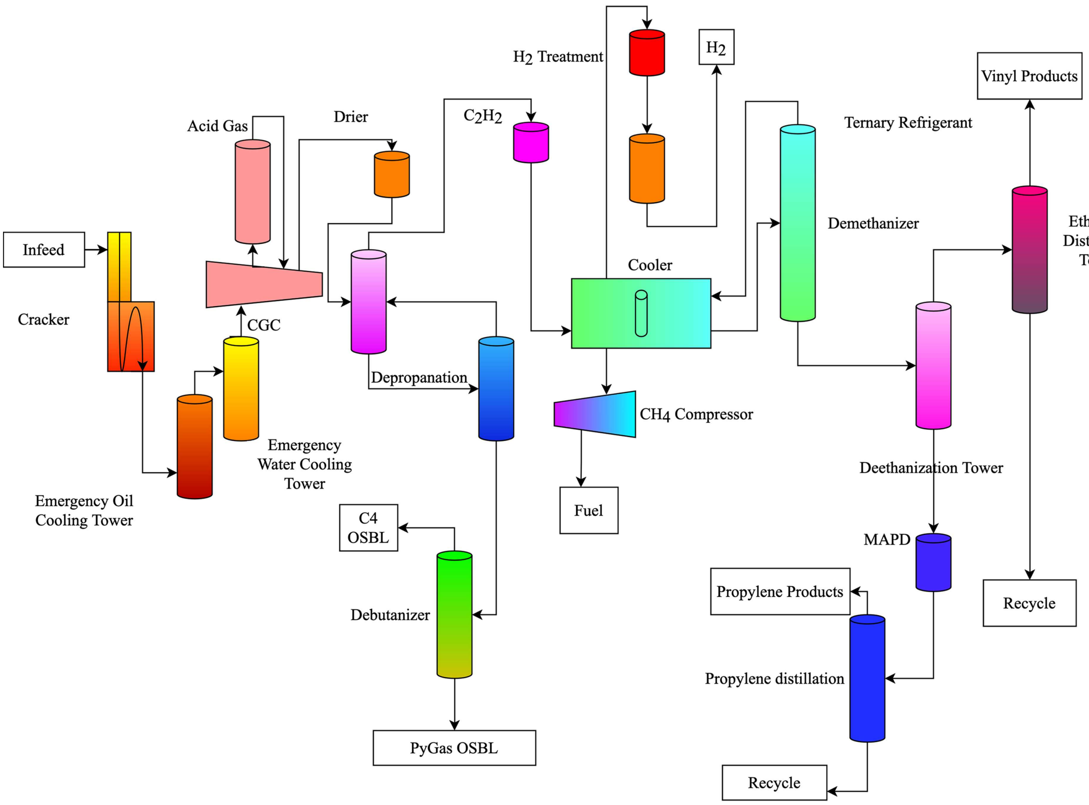

Figure 12 (Color online) Flow chart of the ethylene cracking process.

The optimization is subject to multiple operational constraints, including bounds on the steam-to-hydrocarbon ratio (DSR), naphtha feed flow rate (Q_{naphtha}), cracker outlet temperature (COT), cracking cycle length (Day_{crack}), minimum daily ethylene yield (P_{ethy}), and maximum tube wall temperature (T_w). A key terminal constraint is related to coke thickness, where cracking is stopped when the inner tube diameter reduction reaches one-quarter of the original diameter,

$$d_i \leqslant \frac{1}{4}d, \tag{11}$$

where d_i is the inner diameter of the cracking tube on the i-th day. The initial operating parameters for the cracking furnace model [45] are listed in Table D1 of Appendix D. A schematic of the ethylene cracking process and the furnace structure is provided in Figures D1 and D2 of Appendix D, respectively.

7.2 Case study

Two operational scenarios are compared, namely constant temperature operation and fixed-cycle temperature regulation optimized by the ASPSIOA. In the constant temperature case, the COT is maintained at 1122 K. In the optimization case, the ASPSIOA searches for the best temperature adjustment profile within the allowable range of [1112 K, 1142 K] at 7-day intervals. All other input parameters are kept identical between the two scenarios.

The optimization using the ASPSIOA required approximately 4 h of computation time on an Intel Xeon 16-core processor, which represents only about 0.24% of one actual production cycle. Given that the optimization result can be applied over multiple production cycles, this computational cost is negligible, especially when compared to traditional trial-and-error tuning methods that may require 2–3 full cycles for validation.

The comparative results of ethylene yield before and after optimization are presented in Table 7. The results clearly show that the fixed-cycle temperature regulation strategy found by the ASPSIOA leads to the highest increase in ethylene yield (1.5410%), outperforming all other competitor algorithms.

The optimized temperature profile and the corresponding ethylene yield trend over the cracking days are plotted in Figures 13 and 14. Figure 14 illustrates that under the ASPSIOA-optimized strategy, the decline in ethylene yield over time is significantly smoother and less severe compared to the constant temperature operation. This demonstrates the effectiveness of periodic temperature adjustments in mitigating the negative impact of coking.

Table 7 Comparison of the ethylene optimization yield results for the ethylene cracking furnaces.

	WOA	PSO	ABC	SA	HS	ASPSIOA
Before optimization	28.6208	28.6209	28.7209	28.6202	28.6046	28.4173
After optimization	28.8305	28.7207	–	28.8284	28.7924	28.8621
Increase rate (%)	0.7274	0.3475	–	0.7222	0.6565	1.5410

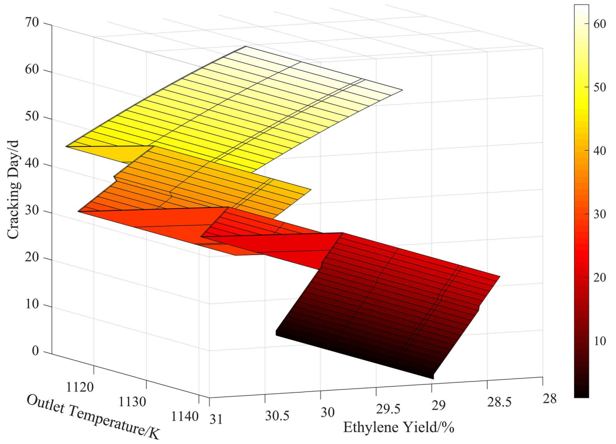

Figure 13 (Color online) Outlet temperature and ethylene yield over cracking days.

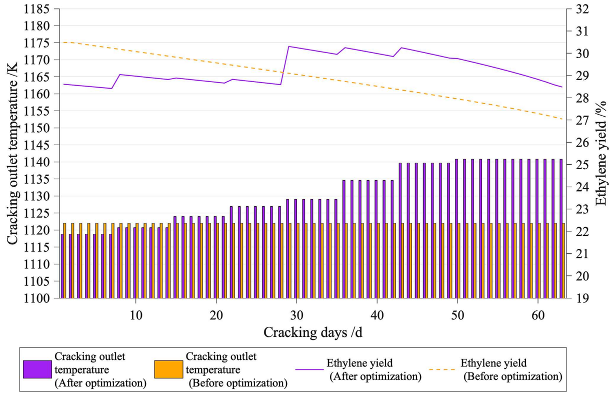

Figure 14 (Color online) Comparison of the ethylene yield and cracking outlet temperature before and after optimization.

The proposed fixed-cycle strategy offers several practical advantages. First, a simple 7-day interval is easier for plant operators to implement and manage compared to complex dynamic controllers. Second, it aligns well with typical plant maintenance schedules, avoiding undue mechanical stress from frequent, large adjustments. Third, as evidenced in Figure 14, the proposed method reduces yield fluctuation by approximately 62%, which is crucial for stabilizing downstream processes. The 1.54% yield improvement translates to an estimated annual production increase of 15400 tons for a million-ton-scale ethylene plant, demonstrating substantial economic benefit and a strong competitive edge for the ASPSIOA-based optimization approach.

8 Conclusion

This paper proposes a novel ASPSIOA, which integrates several key strategies, including an inertial weight, adaptive exploration and exploitation factors, and a small-family group mechanism with elite reverse learning, to enhance overall optimization performance.

The efficacy of the ASPSIOA is rigorously evaluated through an extensive experimental study. The proposed algorithm demonstrates strong global exploration capability, as evidenced by its superior results on multimodal benchmark functions. Its local exploitation strength is confirmed by excellent performance on unimodal functions. Furthermore, the ASPSIOA exhibits a remarkable ability to avoid local optima and handle complex landscapes, which is validated by its success on hybrid and composition functions. Convergence curve analysis indicates that the ASPSIOA possesses a fast initial convergence rate coupled with precise refinement in later stages.

Comparative studies against six established metaheuristic algorithms including the PSO, the GA, the SA, the HS, the ABC, and the WOA confirm the competitiveness of the ASPSIOA. The Wilcoxon signed-rank test provides statistical validation of its superiority, yielding significant p-values as low as 1.73×10^{-6}. Beyond standard benchmarks, ASPSIOA is successfully applied to three semi-realistic constrained engineering problems and a challenging real-world industrial application, namely the optimization of an ethylene cracking furnace operation. By identifying an optimal fixed-cycle temperature regulation strategy to counteract coking effects, the ASPSIOA achieves a 1.541% increase in ethylene yield, demonstrating significant practical utility and economic potential.

This study acknowledges certain limitations rooted in industrial practice. First, the temperature regulation cycle is fixed at 7 days to comply with plant safety and operational norms. Second, detailed energy consumption data are unavailable due to confidentiality policies, which constrains a more comprehensive economic analysis. Future work will focus on constructing a more complete energy consumption model via digital twin technology. Additionally, the development of a multi-objective version of ASPSIOA and its application to other domains such as photovoltaic power generation and catalytic cracking processes are planned directions.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62373035, 62422303) and Xinjiang Uygur Autonomous Region Key R&D Project, China (Grant No. 2023B01031-3).

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chen Z, Li S, Khan A T, et al. Competition of tribes and cooperation of members algorithm: an evolutionary computation approach for model free optimization. Expert Syst Appl, 2025, 265: 125908

2 Tapia-Avitia J M, Cruz-Duarte J M, Amaya I, et al. Analysing hyper-heuristics based on neural networks for the automatic design of population-based metaheuristics in continuous optimisation problems. Swarm Evolary Computation, 2024, 89: 101616

3 Rajwar K, Deep K, Das S. An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artif Intell Rev, 2023, 56: 13187–13257

4 Peng Z J, Liu X Y, Liu X, et al. Performance analysis of active RIS-aided multi-pair full-duplex communications with spatial correlation and imperfect CSI. Sci China Inf Sci, 2023, 66: 192304

5 Wang Y, He Z, Xie S, et al. Explainable prediction of surface roughness in multi-jet polishing based on ensemble regression and differential evolution method. Expert Syst Appl, 2024, 249: 123578

6 Mei Q C, She J H, Liu Z T, et al. Estimation and compensation of periodic disturbance using internal-model-based equivalent-input-disturbance approach. Sci China Inf Sci, 2022, 65: 182205

7 Tan Y, Ouyang J, Zhang Z, et al. Path planning for spot welding robots based on improved ant colony algorithm. Robotica, 2023, 41: 926–938

8 Zhang R, Yu H, Gao K, et al. A Q-learning based artificial bee colony algorithm for solving surgery scheduling problems with setup time. Swarm Evolary Computation, 2024, 90: 101686

9 Shen F, Zhang W, Cao Y, et al. Parameter identification of photovoltaic discrete-time equivalent model using the bat algorithm. Energy Rep, 2023, 9: 449–458

10 Jaisiva S, Prabaakaran K, Kumar C, et al. A novel solution for the optimal reactive power dispatch problem using an artificial neural network integrated with the firefly optimization algorithm. Front Energy Res, 2023, 11: 1310010

11 Zhang T, Wei M, Gao X. Modeling an optimal environmentally friendly energy-saving flexible workshop. Appl Sci, 2023, 13: 11896

12 Carbas S, Tunca O. Numerical design optimization of real size complex steel space frame structures using a novel adaptive cuckoo search method. Numer Meth Eng, 2024, 125: e7386

13 Yang J, Ding T, Deng Q, et al. A multi-UAV indoor air real-time detection and gas source localization method based on improved teaching-learning-based optimization. Atmos Environ, 2024, 318: 120200

14 Yu X, Hu Z, Wang X, et al. Ranking teaching-learning-based optimization algorithm to estimate the parameters of solar models. Eng Appl Artif Intell, 2024, 123: 106225

15 Li X, Li X, Yang G. A novelty harmony search algorithm of image segmentation for multilevel thresholding using learning experience and search space constraints. Multimed Tools Appl, 2023, 82: 703–723

16 Majid N. Collaborative beamforming optimization using imperialist competitive algorithm. Electrotechnical Rev, 2023, 1: 126–131

17 Liu C, Ge S, Zhang Y. Identifying the cardinality-constrained critical nodes with a hybrid evolutionary algorithm. Inf Sci, 2023, 642: 119140

18 Liu X, Liu L, Jiang T. A self-learning interior search algorithm based on reinforcement learning for energy-aware job shop scheduling problem with outsourcing option. J Intell Fuzzy Syst, 2023, 44: 10085–10100

19 Zhang X, Mo T, Zhang Y. Optimization of storage location assignment for non-traditional layout warehouses based on the firework algorithm. Sustainability, 2023, 15: 10242

20 Sharifian Y, Abdi H. Solving multi-area economic dispatch problem using hybrid exchange market algorithm with grasshopper optimization algorithm. Energy, 2023, 267: 126550

21 Gao J, Lv Y, Liu M, et al. Improving simulated annealing for clique partitioning problems. J Artif Intell Res, 2022, 74: 1485–1513

22 Huang F, Zhang H, Wu Q, et al. An optimal model and application of hydraulic structure regulation to improve water quality in plain river networks. Water, 2023, 15: 4297

23 Roy A, Verma V, Gampa S R, et al. Planning of distribution system considering residential roof top photovoltaic systems, distributed generations and shunt capacitors using gravitational search algorithm. Comput Electrical Eng, 2023, 111: 108960

24 Periakaruppan S, Shanmugapriya N, Sivan R. Retracted: self-attention generative adversarial capsule network optimized with atomic orbital search algorithm based sentiment analysis for online product recommendation. J Intell Fuzzy Syst, 2023, 44: 9347–9362

25 Wolpert D H, Macready W G. No free lunch theorems for optimization. IEEE Trans Evol Computat, 1997, 1: 67–82

26 Liang J, Suganthan P, Deb K. Novel composition test functions for numerical global optimization. In: Proceedings 2005 IEEE Swarm Intelligence Symposium, Pasadena, 2005. 68–75

27 Yao X, Liu Y, Lin G M. Evolutionary programming made faster. IEEE Trans Evol Computat, 1999, 3: 82–102

28 Digalakis J G, Margaritis K G. On benchmarking functions for genetic algorithms. Int J Comput Math, 2001, 77: 481–506

29 Jamil M, Yang X, Zepernick H. Test functions for global optimization: a comprehensive survey. Swarm Intell Bio-Inspired Comput, 2013, 2013: 193–222

30 Yang X S. Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput, 2010, 2: 78

31 Suganthan P, Hansen N, Liang J, et al. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. Nat Comput, 2005, 2005: 341–357

32 Katoch S, Chauhan S S, Kumar V. A review on genetic algorithm: past, present, and future. Multimed Tools Appl, 2021, 80: 8091–8126

33 Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Software, 2016, 95: 51–67

34 Mahmoodabadi M J, Rasekh M, Yahyapour M. Tree optimization algorithm (TOA): a novel metaheuristic approach for solving mathematical test functions and engineering problems. Evol Intel, 2023, 16: 1325–1338

35 Kumar A, Wu G, Ali M Z, et al. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm Evolary Computation, 2020, 56: 100693

36 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, 2002, 191: 1245–1287

37 Sandgren E. Nonlinear integer and discrete programming in mechanical design. In: Proceedings of the 14th Design Automation Conference, Kissimmee, 1988. 95–105

38 Gupta S, Tiwari R, Nair S B. Multi-objective design optimisation of rolling bearings using genetic algorithms. Mechanism Machine Theor, 2007, 42: 1418–1443

39 Sigmund O. A 99 line topology optimization code written in Matlab. Struct Multidisc Optim, 2001, 21: 120–127

40 Wang Y, Geng Z, Hu X, et al. Production prediction and energy saving of complex industrial processes using multiscale variable dynamic interaction information extraction network integrating regressor. Energy, 2025, 326: 136377

41 Han Y, Ji W, Liu L, et al. Impact of the national economy and air pollutant emissions on Chinese energy mix: evidence from an SBMDEA-TOPSIS model. Energy, 2025, 325: 136098

42 Zhao C, Liu C, Xu Q. Cyclic scheduling for ethylene cracking furnace system with consideration of secondary ethane cracking. Ind Eng Chem Res, 2010, 49: 5765–5774

43 Cui D, Peng Z, Huang H, et al. Next-generation 5G fusion-based intelligent health-monitoring platform for ethylene cracking furnace tube. Math Biosci Eng, 2022, 19: 9168–9199

44 Lin X, Zhao L, Du W, et al. Data-driven modeling and cyclic scheduling for ethylene cracking furnace system with inventory constraints. Ind Eng Chem Res, 2021, 60: 3687–3698

45 Wang X, Geng Z, Chen L, et al. Improved dragonfly optimization algorithm based on quantum behavior for multi-objective optimization of ethylene cracking furnace. Swarm Evolary Computation, 2024, 88: 101607