Expert Systems With Applications 302 (2026) 130489

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Sparse-driven auxiliary mask regulation and hierarchical action selection for large-scale multi-objective optimization

Shuzhao Pang [a], Qingke Zhang [a,*], Zhi-Hui Zhan [b], Junqing Li [c], Huaxiang Zhang [a]

[a] School of Information Science and Engineering, Shandong Normal University, Jinan, 250358, China
[b] College Of Artificial Intelligence, Nankai University, Tianjin, 300350, China
[c] School of Mathematics, Yunnan Normal University, Kunming, 650500, China

ARTICLE INFO

Keywords:
Multi-objective optimization
Sparse optimization
Deep reinforcement learning
Evolutionary computation.

ABSTRACT

This paper addresses the optimization challenges of large-scale sparse multi-objective problems, which are characterized by high-dimensional decision variables, heterogeneous variable bounds, and complex structural dependencies. To tackle these issues, we propose an improved algorithm based on an automated guiding vector selection-based evolutionary algorithm (AGSEA), termed SMAF, which integrates structure-aware sampling and dynamic guidance. Specifically, structured sampling is employed to exploit latent variable relationships, with a fixed-value evaluation scheme for performance assessment. A dynamically weighted fitness evaluation mechanism is introduced to enrich the action space, while the sparsity levels of individuals are adaptively adjusted according to feedback from the sparsity patterns of non-dominated solutions. Comprehensive comparisons with seven state-of-the-art multi-objective evolutionary algorithms are conducted on benchmark problems and four real-world applications. The results show that SMAF consistently achieves competitive convergence and stability across the bi-objective test scenarios, with at least about a 23.6 % improvement in Inverted Generational Distance (IGD) over the comparison algorithms across 24 test instances. These results confirm the effectiveness and adaptability of SMAF for high-dimensional sparse multi-objective optimization.

1. Introduction

In recent years, with the rapid development of fields such as financial modeling, bioinformatics, and artificial intelligence, decision makers are increasingly required to optimize multiple conflicting objectives simultaneously Sharma and Kumar (2022). For instance, constructing an investment portfolio requires balancing return maximization with risk minimization. As a result, multi-objective optimization (MOO) has become a central research topic in the optimization community He et al. (2025), Wang et al. (2023b).

However, while traditional MOO algorithms have achieved considerable success on low-dimensional problems, they often struggle to scale with the growing complexity of real-world applications Shang et al. (2025). Modern systems, such as large-scale biological datasets and deep learning models, may involve thousands of variables, which substantially increases the difficulty of optimization by introducing multiple conflicting objectives as well as severe computational and convergence challenges Liu and Liu (2024), Tian et al. (2024a), Wu et al. (2023).

Beyond high dimensionality, sparsity introduces an additional layer of complexity. In many real-world tasks, such as feature selection, a large portion of decision variables are irrelevant or only weakly correlated with the objectives. Effectively identifying and leveraging such sparse structures is crucial, as it can reduce computational costs, enhance interpretability, and improve generalization performance Tian et al. (2024b).

Motivated by these challenges, large-scale sparse multi-objective optimization problems (sparse LSMOPs) have attracted growing attention Li et al. (2023), Sun and Jiang (2024), Zhang et al. (2023a). This research direction aims to integrate sparse representation techniques with advanced MOO strategies, in order to efficiently solve optimization tasks in extremely high-dimensional spaces.

To address the scalability issue, some recent works have explored transfer learning and deep reinforcement learning to enhance algorithm adaptability across different problem instances. Among them, the AGSEA algorithm Shao et al. (2024) introduces a deep reinforcement learning framework for guiding vector selection, which helps improve convergence and diversity in sparse MOPs.

However, AGSEA still encounters several challenges in large-scale sparse multi-objective optimization, particularly in high-dimensional and non-uniform variable spaces. The evaluation of single-variable

* Corresponding author.
E-mail addresses: shuzhao.pang@stu.sdnu.edu.cn (S. Pang), tsingke@sdnu.edu.cn (Q. Zhang), zhanapollo@163.com (Z.-H. Zhan), lijunqing@ynnu.edu.cn (J. Li), huaxzhang@sdnu.edu.cn (H. Zhang).

https://doi.org/10.1016/j.eswa.2025.130489
Received 10 August 2025; Received in revised form 5 November 2025; Accepted 16 November 2025
Available online 3 December 2025
0957-4174/© 2025 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

contributions can be influenced by the randomness of sampled values, which may obscure the genuine effect of individual variables and reduce the stability of variable importance estimation. Moreover, the auxiliary mask generation mechanism–where the mask is represented as a binary matrix indicating whether a decision variable is activated (1) or deactivated (0) during the evolutionary process–lacks adaptive regulation based on optimization feedback. This may lead to imbalanced exploration of sparse regions and insufficient utilization of variable-level sparsity interval information. These limitations restrict the algorithm's ability to fully capture sparse structures and maintain stable search performance in practical LSMOP scenarios, highlighting the need for further methodological enhancements Peng and Ishibuchi (2020), Wang et al. (2023a), Zou et al. (2023).

Based on the above analysis, this paper systematically improves the deep reinforcement learning module within the AGSEA algorithm. Specifically, it includes constructing a more expressive state representation, designing a hierarchical action decision process to achieve effective partitioning and dynamic fusion of the action space, and introducing an auxiliary mask regulation mechanism guided by sparsity intervals. These improvements aim to enhance the stability and generalization capability of the algorithm in complex sparse multi-objective optimization problems. The main contributions of this paper are summarized as follows:

- A systematic variable importance assessment framework is proposed, integrating multi-level interval stratified sampling with fixed-value control-variable evaluation. Specifically, the feasible range of each decision variable is partitioned into three representative subintervals (low, medium, high), and a fixed value–typically the midpoint–is selected from each for evaluation. By activating one variable at a time while keeping others inactive, the individual contribution of each variable is isolated and assessed under controlled conditions. These importance scores serve as prior knowledge to inform structured variable grouping and guide the construction of sparse decoding structures in subsequent optimization stages, thereby ensuring a fair, consistent, and reproducible evaluation across all variables.
- A dynamic weighted fusion fitness evaluation method is proposed to dynamically adjust the weights between uniform sampling fitness and structured sampling fitness throughout the optimization process. In combination with a Softmax action selection strategy and an auxiliary mask regulation mechanism based on sparsity intervals of elite solutions on the non-dominated front, this approach computes action probability distributions and adaptively adjusts mask sparsity.
- The proposed SMAF algorithm is evaluated on eight benchmark SMOP problems and four real-world applications, compared with several state-of-the-art multi-objective evolutionary algorithms. Experimental results indicate that SMAF performs competitively in terms of solution quality and convergence speed, achieving comparable or superior performance in most cases.

The remainder of this paper is organized as follows. Section 2 reviews fundamental concepts of multi-objective optimization problems and surveys existing sparse LSMOPs. Section 3 provides a detailed description of the proposed SMAF algorithm. Section 4 presents and discusses the experimental results. Section 5 reports ablation studies analyzing the contributions of different components. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Related works

2.1. Background of sparse LSMOPs

Multi-Objective Optimization (MOO) is dedicated to optimizing multiple conflicting objective functions simultaneously, and has a wide range of applications in engineering design, resource scheduling and system modeling Hasanzadeh et al. (2023), He et al. (2020), Ma et al. (2023). Compared with single-objective optimization, MOO no longer seeks a single globally optimal solution, but focuses on the solution set that reflects the trade-off relationship between the objectives, i.e., the Pareto Optimal Set and its mapping in the objective space - the Pareto Front Verma et al. (2021).

A typical multi-objective optimization problem can be formally defined as shown in Eq. (1), where $\mathbf{x} \in \mathbb{R}^D$ denotes the decision vector, and $\mathbf{F}(\mathbf{x})$ represents the objective function vector with m objectives. Ω is the feasible domain of the decision variable Khishe et al. (2023).

$$\begin{cases} \text{minimize} \quad \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T \\ \text{subject to} \quad \mathbf{x} \in \Omega \end{cases} \tag{1}$$

LSMOPs have received increasing attention from researchers as problem sizes and decision variable dimensionalities increase significantly Liu et al. (2024), Xu et al. (2022), Yang et al. (2021). To address the challenges of low search efficiency and susceptibility to local optima in high-dimensional spaces, traditional optimization algorithms have incorporated techniques such as dimensionality reduction, variable grouping, and novel search strategy-based algorithms to enhance convergence performance and solution diversity Tian et al. (2021), Wang et al. (2020, 2019).

Dimensionality reduction methods aim to reduce the search dimension of optimization algorithms while preserving the essential information of the original decision space Tian et al. (2020a). Among these, techniques based on deep learning and pattern mining have gained significant attention due to their strong capability to represent complex high-dimensional data. Taking MOEA/PSL Tian et al. (2020b) as an example, it efficiently achieves dimensionality reduction by learning a Pareto optimal subspace. The algorithm represents each solution using a binary vector indicating the sparsity pattern of variables and a real-valued vector for the non-zero variable values. It then employs Restricted Boltzmann Machines (RBM) Fischer and Igel (2012) and Denoising Auto-Encoders (DAE) Xue et al. (2012) to extract and compress features from the binary and real-valued vectors, respectively, jointly constructing a low-dimensional subspace. Throughout the evolutionary process, MOEA/PSL balances exploration and exploitation by dynamically adjusting the search ratio between the reduced subspace and the original space, as well as adaptively tuning the model complexity.

R.H.M. Coppens *et al.* proposed a framework that integrates multi-objective evolutionary algorithms (MOEAs) with deep reinforcement learning (DRL) Li et al. (2025) for solving multi-objective optimization problems Coppens et al. (2022). In this framework, DRL policies are employed to dynamically adjust the mutation intensity and probability parameters within MOEAs, thereby enhancing the optimization performance. Hai Long Tran *et al.* proposed the MOPDERL Tran et al. (2023) framework, which is a two-stage multi-objective evolutionary reinforcement learning approach designed for continuous robot control problems. By combining the robustness of evolutionary algorithms with the efficiency of reinforcement learning, this framework effectively addresses multi-objective continuous control tasks. It should be noted that these approaches, while effective for general multi-objective optimization problems, lack specific mechanisms to handle high-dimensional sparse decision spaces. Therefore, they are not included in the comparison with large-scale sparse multi-objective algorithms considered in this study.

In addition to exploiting the sparse structure, optimizing the search mechanism is also crucial for enhancing the performance of large-scale multi-objective optimization algorithms. SparseEA Tian et al. (2019) efficiently searches for sparse Pareto-optimal solutions by integrating a sparsity-aware population initialization strategy with hybrid genetic operators. Each solution comprises a binary mask vector and a real-valued decision vector, jointly encoding the sparsity structure and decision values. Variable importance, derived from non-dominated sorting, guides the initialization to retain key variables with a higher probability of being activated. During evolution, the mask vectors undergo score-guided binary variation to preserve sparsity, while the decision vectors are evolved via simulated binary crossover and polynomial mutation. This design enables SparseEA to effectively maintain solution sparsity

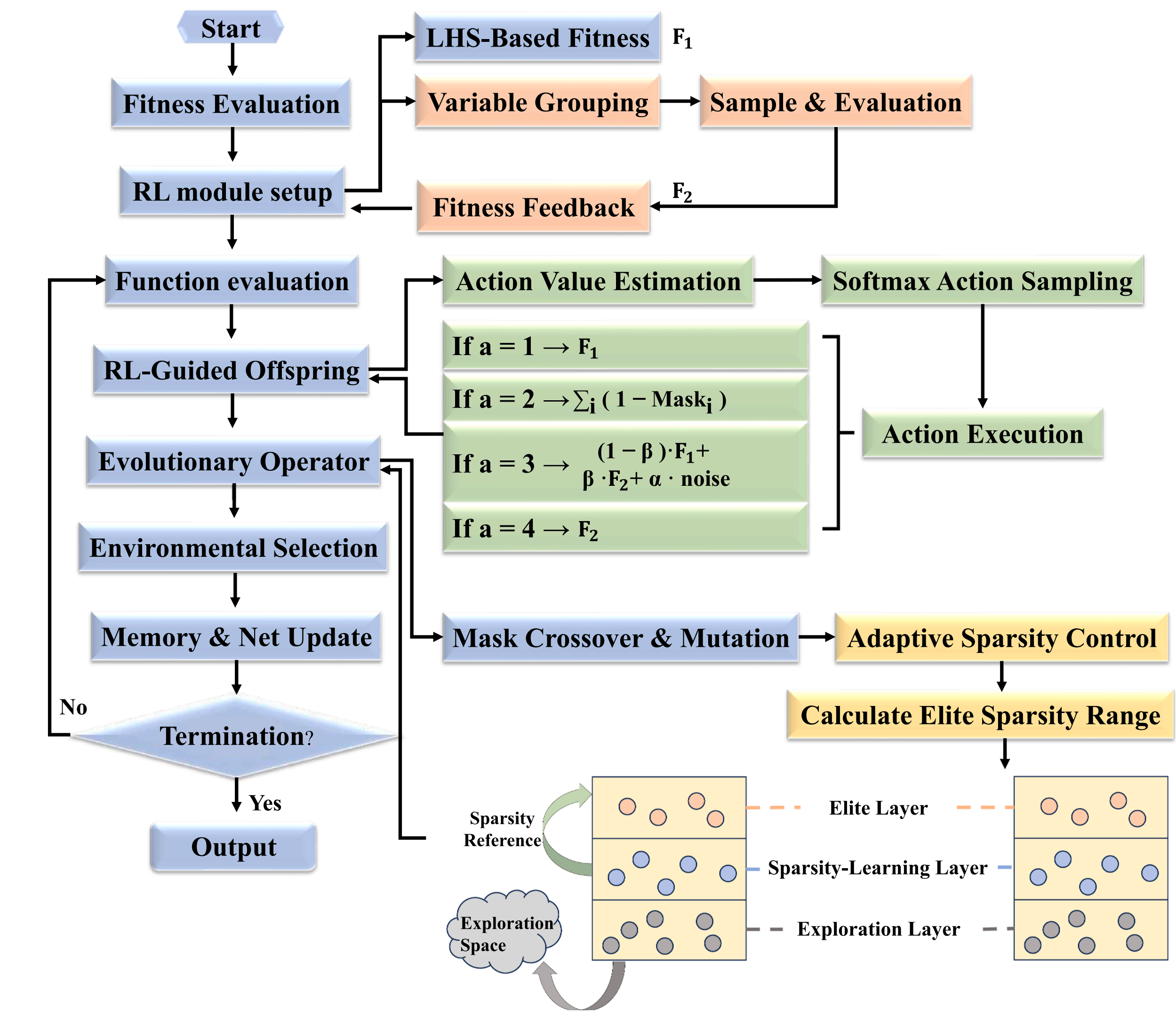

Fig. 1. The framework of the proposed algorithm.

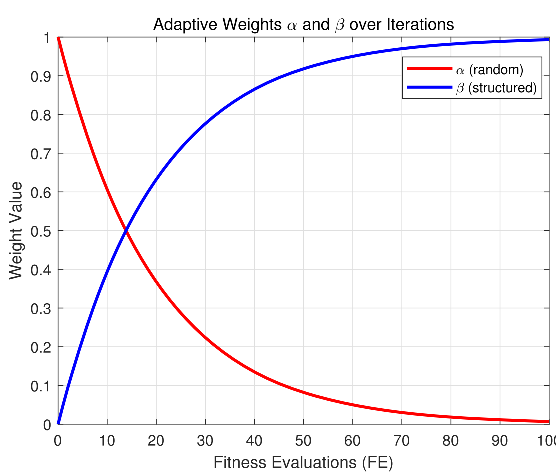

Fig. 2. Adaptive weights α (red) and β (blue) over iterations, showing the transition from exploration to exploitation.

while exploring a diverse set of high-quality trade-offs, making it well-suited for large-scale sparse multi-objective optimization problems Ding et al. (2023).

Although methods such as SparseEA have made significant progress in modeling and exploiting sparse structures, they still face challenges when dealing with complex problems characterized by non-uniform variable distributions or dynamically changing sparsity Zhang et al. (2024). Specifically, such methods often fail to promptly capture variations in variable importance, which limits both the diversity of the obtained solutions and the efficiency of convergence. To address this, some recent studies have attempted to incorporate reinforcement learning mechanisms Raju and Mothku (2023), Siraskar et al. (2023), Zhao et al. (2023) into sparse multi-objective optimization, aiming to achieve dynamic learning and self-adaptive control of search strategies Chu et al. (2024).

For instance, the AGSEA algorithm employs deep reinforcement learning to adaptively select guidance vectors by mapping current population features as states, improvement in convergence and diversity as rewards, and candidate guidance vectors as actions. This adaptive strategy enables AGSEA to effectively guide the search process in sparse LSMOPs, showing competitive performance on various benchmarks.

While AGSEA achieves competitive performance, there remains room for further enhancement, for instance in enriching the action space, adopting more adaptive mask perturbation strategies, and so on. Motivated by these challenges, this paper proposes a novel deep reinforcement learning framework that incorporates sparsity-aware mechanisms and a multi-level action selection strategy Tian et al. (2022a).

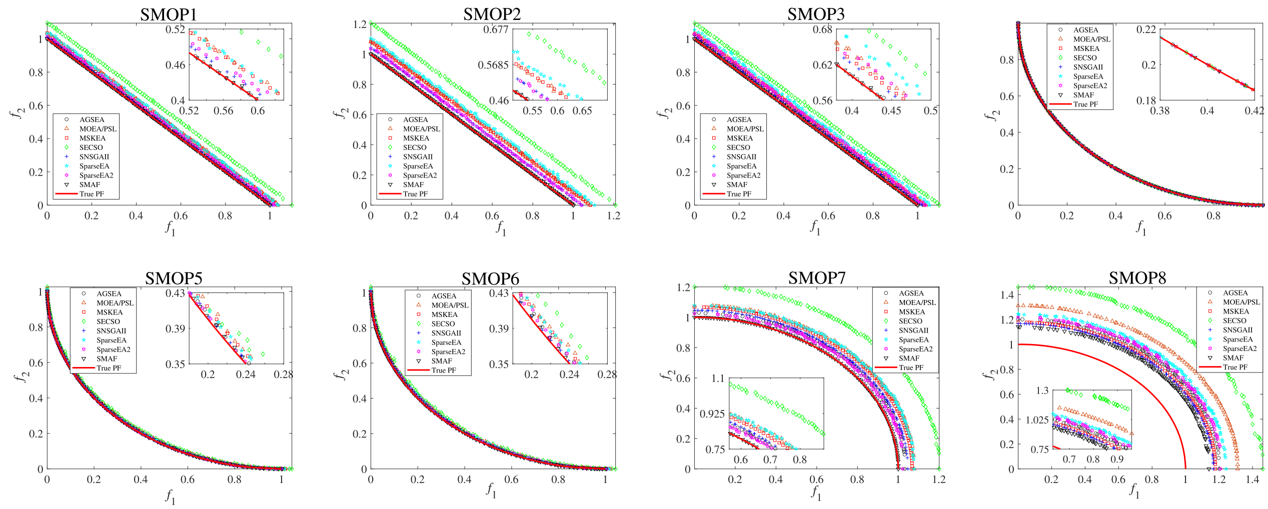

Fig. 3. Populations with the median IGD obtained by SMAF and its competitors on the bi-objective SMOP1-SMOP8 problems with 500 decision variables.

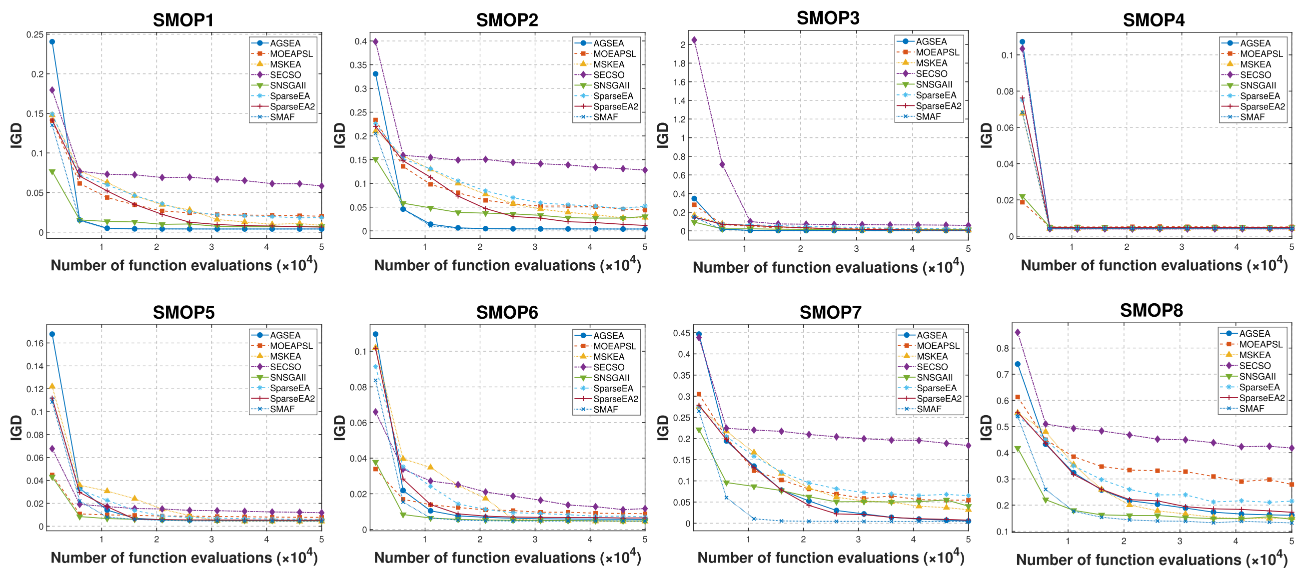

Fig. 4. The IGD values (median) obtained by SMAF and its comparison algorithms on the bi-objective SMOP1-SMOP8, with 500 decision variables under the number maximum of fitness evaluations varying from 1000 to 50,000.

The proposed approach systematically extends AGSEA by redesigning its action modeling and mask perturbation components, with the goal of improving robustness and generalization in large-scale sparse multi-objective optimization Ren et al. (2023).

2.2. Motivation of this work

In sparse LSMOPs, conventional frameworks such as MOEA/D, NSGA-II variants, and cooperative co-evolutionary methods lack explicit mechanisms to identify and exploit variable sparsity, making them ineffective for generating high-quality sparse solutions.

AGSEA provides a natural baseline for sparse optimization by explicitly modeling variable importance through its mask mechanism, making it well-suited for high-dimensional and sparse problems. Its mask-based design also facilitates learning-based enhancements: mask modification and operator selection can be interpreted as actions in a reinforcement learning framework, while the current population distribution, variable contribution estimates, and mask sparsity information can serve as states. This structure allows dynamic adjustment of search strategies and the seamless integration of new evaluation methods or adaptive sparsity control without disrupting the core framework, a flexibility that conventional MOEA/D or NSGA-II lack.

Despite these strengths, AGSEA has several limitations. First, the contribution of individual variables can be distorted by random values during evaluation, undermining the stability of variable importance estimation. Second, its mask sparsity control lacks adaptive feedback, potentially causing uneven exploration of sparse regions. These issues are particularly pronounced in high-dimensional and non-uniform variable spaces.

To overcome these limitations, the proposed algorithm introduces several enhancements absent in AGSEA: (1) Stratified fixed-value control variable evaluation – partitioning the decision space and evaluating each variable under fixed conditions to provide more stable contribution estimates; (2) Dynamic action guidance – combining multi-source fitness information and population feedback to adaptively adjust mask sparsity and operator selection; (3) Adaptive sparsity control – guiding offspring mask sparsity based on feedback from current non-dominated solutions, improving exploration in sparse regions.

Compared to the original AGSEA, these improvements enhance the stability of variable contribution estimation, enable adaptive sparsity adjustment, and leverage dynamic guidance from population feedback, resulting in faster convergence and more reliable identification of sparse solutions. Subsequent experimental results validate the effectiveness of these enhancements.

3. The proposed algorithm

Fig. 1 illustrates the overall framework of the proposed algorithm, SMAF, which is introduced in Section 3. It comprises key modules such as initialization, state evaluation, policy updating, and result output, together reflecting the core process of the algorithm and the interactions among its components. Detailed implementations are provided in the subsequent sections.

3.1. *Fixed-Value control-variable evaluation strategy in structured decision spaces*

In large-scale sparse multi-objective optimization problems, the convergence efficiency and solution quality of an algorithm largely depend on the accurate identification of key decision variables Hu et al. (2025). In high-dimensional decision spaces, a large number of redundant or ineffective variables not only fail to contribute positively to the objectives, but may also mislead the search toward low-quality regions and interfere with the identification of the Pareto-optimal subspace. Therefore, constructing guidance vectors that reveal the importance structure of decision variables in the early stages of optimization, and leveraging them to guide the generation of sparse decoding structures (e.g., masks), is a critical issue.

The AGSEA algorithm employs an evaluation approach based on a uniform sampling strategy to initialize the population by broadly covering the decision space. Candidate solutions are generated using Latin Hypercube Sampling (LHS) McKay et al. (2000), and a masking mechanism is incorporated to assess the contribution of each decision variable dimension-wise. The main advantage of this approach lies in its computational efficiency and relatively uniform spatial coverage. It should be noted that this evaluation approach, which corresponds to the component F_1 in Fig. 1, inherently involves some randomness. For real-valued decision variables, the activated values are generated via uniform or Latin Hypercube Sampling within their feasible ranges. Consequently, each run produces different sampled values for the same variable, leading to variations in the computed Fitness vector and, hence, a non-deterministic ranking of variable contributions across runs.

To address this issue and obtain systematic assessments of variable contributions, a fixed-value, control-variable sampling strategy is employed. Specifically, each decision variable x_i is examined within its feasible interval $[l_i, u_i]$, which is divided into $k = 3$ equal subintervals to enable systematic exploration across low, medium, and high regions. Within each subinterval, a representative candidate value is placed at the midpoint, as defined serving as the fixed value for controlled evaluation. If the midpoint coincides with zero, it is shifted to the upper bound u_i to avoid ineffective zero-valued variables.

For each subinterval, sparse individuals are constructed by activating only the selected variable while keeping others inactive. These individuals are then evaluated using the problem's objective and constraint functions, yielding preliminary importance scores for each variable position. By activating one variable at a time and combining fixed values with interval-based sampling, this strategy ensures that each position is assessed under controlled and comparable conditions. These importance scores provide reliable prior knowledge to guide the generation of sparse decoding structures in subsequent optimization stages.

The representative point of the j-th subinterval for decision variable $x_i \in [l_i, u_i]$ is computed as (see Eq. (2)):

$$\hat{x}_i^{(j)} = l_i + \frac{2j-1}{2k}(u_i - l_i), \quad j = 1, \dots, k, \tag{2}$$

where k denotes the number of subintervals and j indexes the subinterval. The fraction $\frac{2j-1}{2k}$ represents the normalized midpoint within the

Table 1
Parameter settings of eight benchmark SMOPs and four practical SMOPs.

Benchmark problem	Type of variables	No. of variables (D)	No. of objective M	Sparsity of Pareto optimal solutions		
SMOP1–SMOP8	Real	500 1000 2000	2, 3	0.1		
Sparse signal reconstruction problem	**Type of variables**	**No. of variables (D)**	**Dataset**	**Length of signal**	**Number of observations**	**Sparsity of signal**
SR1	Real	1024	Synthetic Liang et al. (2018)	1024	480	260
SR2		2048	Synthetic Liang et al. (2018)	2048	960	520
SR3		5120	Synthetic Liang et al. (2018)	5120	2400	1300
Feature selection problem	**Type of variables**	**No. of variables (D)**	**Dataset**	**No. of samples**	**No. of features**	**No. of classes**
FS1	Binary	166	MUSK1[a]	476	166	2
FS2		256	Semeion Handwritten Digit[a]	1593	256	10
FS3		310	LSVT Voice rehabilitation[a]	126	310	2
Critical node detection	**Type of variables**	**No. of variables (D)**	**Dataset**	**No. of nodes**	**No. of edges**	
CN1	Binary	234	GD(A99)[b]	234	154	
CN2		311	GD(A01)[b]	311	640	
CN3		452	GD(C97)[b]	452	460	
Pattern mining problem	**Type of variables**	**No. of variables (D)**	**Dataset**	**No. of transactions**	**No. of items**	**Avg. length of variables variables transactions**
PM1	Binary	500	Synthetic Agrawal and Srikant (1994)	10 000	500	50
PM2		1000	Synthetic Agrawal and Srikant (1994)	10 000	1000	50
PM3		2000	Synthetic Agrawal and Srikant (1994)	10 000	2000	50

[a] http://archive.ics.uci.edu/ml/index.php
[b] http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm

Table 2
The mean and standard deviation of IGD results obtained by SMAF and its comparison algorithms on the SMOP1-SMOP8 bi-objective benchmark problems, where the best result in each row is highlighted. '+', '-', and '≈' indicate that the result by another MOEA is better than, worse than, or comparable to that obtained by SMAF.

Problem	D	MSKEA	S-NSGA-II	SparseEA	SparseEA2	AGSEA	S-ECSO	MOEA/PSL	SMAF
SMOP1	500	2.5564e-2 (4.72e-3) -	1.0869e-2 (3.49e-3) -	2.9223e-2 (3.80e-3) -	1.6024e-2 (3.12e-3) -	4.0921e-3 (9.71e-5) -	6.9429e-2 (2.26e-3) -	2.7206e-2 (4.60e-3) -	**3.9603e-3 (7.05e-5)**
	1000	3.4014e-2 (3.13e-3) -	1.2511e-2 (2.86e-3) -	3.5485e-2 (2.83e-3) -	2.0996e-2 (2.60e-3) -	4.2370e-3 (1.40e-4) -	7.0136e-2 (2.11e-3) -	3.3493e-2 (9.74e-3) -	**4.0478e-3 (6.56e-5)**
	2000	3.8447e-2 (2.68e-3) -	1.6893e-2 (3.82e-3) -	4.1871e-2 (2.19e-3) -	3.2381e-2 (2.52e-3) -	4.3553e-3 (1.51e-4) -	7.0891e-2 (1.08e-3) -	3.7313e-2 (3.61e-3) -	**4.0468e-3 (5.73e-5)**
SMOP2	500	6.2370e-2 (7.91e-3) -	3.3632e-2 (1.09e-2) -	7.6209e-2 (6.92e-3) -	3.4257e-2 (6.66e-3) -	5.0688e-3 (1.70e-3) -	1.4503e-1 (3.58e-3) -	6.0752e-2 (8.00e-3) -	**4.2087e-3 (1.20e-4)**
	1000	7.8099e-2 (6.71e-3) -	5.3309e-2 (1.09e-2) -	8.4786e-2 (5.11e-3) -	4.6753e-2 (4.38e-3) -	5.3630e-3 (1.45e-3) -	1.4807e-1 (3.27e-3) -	6.9187e-2 (7.92e-3) -	**4.4260e-3 (1.29e-4)**
	2000	9.0240e-2 (4.51e-3) -	7.1114e-2 (1.59e-2) -	1.0177e-1 (4.21e-3) -	7.0163e-2 (4.06e-3) -	5.3317e-3 (1.45e-3) -	1.4951e-1 (2.62e-3) -	8.1528e-2 (5.58e-3) -	**4.3656e-3 (1.17e-4)**
SMOP3	500	1.9820e-2 (5.95e-3) -	1.6175e-2 (4.37e-3) -	4.2830e-2 (4.85e-3) -	2.6199e-2 (3.46e-3) -	**4.0174e-3 (2.70e-4)** =	6.9548e-2 (2.60e-3) -	8.5146e-2 (3.11e-1) -	4.0475e-3 (2.52e-4)
	1000	2.7764e-2 (3.52e-3) -	1.9990e-2 (4.20e-3) -	4.7183e-2 (2.70e-3) -	3.2661e-2 (3.41e-3) -	4.0654e-3 (2.77e-4) =	6.9736e-2 (1.80e-3) -	2.9069e-2 (2.52e-3) -	**3.9679e-3 (6.99e-5)**
	2000	3.1913e-2 (2.50e-3) -	2.3802e-2 (4.31e-3) -	5.2630e-2 (2.31e-3) -	4.0633e-2 (2.78e-3) -	4.6204e-3 (1.85e-3) -	7.0506e-2 (1.10e-3) -	3.8596e-2 (5.38e-3) -	**3.9668e-3 (8.76e-5)**
SMOP4	500	4.1370e-3 (5.48e-5) =	4.9182e-3 (2.86e-4) -	4.7385e-3 (2.32e-4) -	4.8676e-3 (2.38e-4) -	4.1406e-3 (8.35e-5) =	**3.9657e-3 (7.47e-5)** +	5.1152e-3 (4.66e-4) -	4.1267e-3 (5.18e-5)
	1000	4.1052e-3 (7.70e-5) =	4.7913e-3 (2.46e-4) -	4.7104e-3 (2.01e-4) -	4.7747e-3 (2.01e-4) -	4.1257e-3 (5.67e-5) =	**3.9616e-3 (7.02e-5)** +	5.1441e-3 (3.09e-4) -	4.1348e-3 (7.80e-5)
	2000	4.1303e-3 (7.18e-5) =	4.7822e-3 (2.82e-4) -	4.7542e-3 (2.08e-4) -	4.7249e-3 (1.96e-4) -	4.1389e-3 (6.58e-5) =	**3.9979e-3 (6.29e-5)** +	5.1933e-3 (3.49e-4) -	4.1250e-3 (6.79e-5)
SMOP5	500	8.8386e-3 (2.98e-3) -	5.4407e-3 (2.70e-4) -	7.9201e-3 (4.70e-4) -	5.6114e-3 (2.11e-4) -	5.3985e-3 (3.68e-4) -	1.3887e-2 (9.91e-4) -	8.8889e-3 (6.24e-4) -	**4.7723e-3 (1.76e-4)**
	1000	1.2147e-2 (3.13e-3) -	5.7365e-3 (1.16e-3) -	7.6816e-3 (4.61e-4) -	5.2968e-3 (2.12e-4) -	4.9631e-3 (1.93e-4) -	1.3165e-2 (1.19e-3) -	9.5585e-3 (4.37e-4) -	**4.6126e-3 (1.32e-4)**
	2000	1.3555e-2 (1.64e-3) -	8.8114e-3 (3.01e-3) -	7.9176e-3 (4.30e-4) -	5.1992e-3 (2.55e-4) -	4.8177e-3 (1.81e-4) -	1.2762e-2 (1.39e-3) -	9.5759e-3 (4.75e-4) -	**4.5341e-3 (9.53e-5)**
SMOP6	500	9.8824e-3 (3.95e-3) -	5.2066e-3 (2.32e-4) -	9.8844e-3 (9.14e-4) -	7.1508e-3 (4.95e-4) -	6.5200e-3 (6.14e-4) -	1.7335e-2 (2.99e-3) -	1.0773e-2 (6.07e-4) -	**4.7780e-3 (1.97e-4)**
	1000	1.4097e-2 (2.94e-3) -	5.2451e-3 (3.05e-4) -	9.7180e-3 (7.17e-4) -	7.0328e-3 (3.41e-4) -	5.9915e-3 (3.32e-4) -	1.6304e-2 (3.77e-3) -	1.2653e-2 (1.20e-3) -	**4.6740e-3 (1.79e-4)**
	2000	1.6583e-2 (1.64e-3) -	5.4371e-3 (3.84e-4) -	1.0261e-2 (6.00e-4) -	6.9669e-3 (2.83e-4) -	5.9466e-3 (2.85e-4) -	1.6655e-2 (3.63e-3) -	1.3370e-2 (5.25e-4) -	**4.6853e-3 (1.51e-4)**
SMOP7	500	6.8955e-2 (1.07e-2) -	5.3537e-2 (1.49e-2) -	8.1179e-2 (1.05e-2) -	3.1083e-2 (6.85e-3) -	3.5130e-2 (9.30e-3) -	2.0447e-1 (6.68e-3) -	1.4150e-1 (1.29e-1) -	**4.5377e-3 (5.10e-4)**
	1000	8.6415e-2 (8.05e-3) -	7.8087e-2 (1.79e-2) -	1.0084e-1 (5.44e-3) -	3.6089e-2 (5.32e-3) -	5.4255e-2 (1.02e-2) -	2.0984e-1 (4.18e-3) -	1.8971e-1 (1.28e-1) -	**4.6113e-3 (3.80e-4)**
	2000	1.0552e-1 (7.06e-3) -	1.1830e-1 (2.52e-2) -	1.2307e-1 (4.89e-3) -	6.3768e-2 (3.79e-3) -	8.7954e-2 (1.14e-2) -	2.0961e-1 (3.86e-3) -	1.7068e-1 (1.10e-1) -	**4.5134e-3 (1.02e-4)**
SMOP8	500	1.8001e-1 (1.70e-2) -	1.6709e-1 (1.90e-2) -	2.4183e-1 (1.77e-2) -	2.1527e-1 (1.38e-2) -	2.0678e-1 (1.66e-2) -	4.6213e-1 (7.88e-3) -	3.1891e-1 (3.51e-2) -	**1.4219e-1 (9.10e-3)**
	1000	2.1920e-1 (1.45e-2) -	1.8154e-1 (1.81e-2) -	2.7566e-1 (1.32e-2) -	2.4190e-1 (1.26e-2) -	2.2842e-1 (1.68e-2) -	4.6355e-1 (8.58e-3) -	4.2558e-1 (2.91e-2) -	**1.4521e-1 (8.28e-3)**
	2000	2.5704e-1 (1.17e-2) -	1.9990e-1 (1.47e-2) -	3.1372e-1 (1.02e-2) -	2.5257e-1 (1.21e-2) -	2.7711e-1 (1.05e-2) -	4.6474e-1 (9.33e-3) -	4.7075e-1 (1.44e-2) -	**1.4931e-1 (3.00e-3)**
+/-/=		0/21/3	0/24/0	0/24/0	0/24/0	0/19/5	3/21/0	0/24/0	

interval. If the computed midpoint equals zero, it can be replaced with a non-zero value, such as the upper bound u_i, to avoid generating ineffective samples.

For each subinterval j, these midpoint values are assigned to the corresponding variables to construct sparse individuals, which are then evaluated using the problem's objective and constraint functions. From these evaluations, preliminary variable importance scores are obtained, providing prior knowledge to guide subsequent optimization.

Compared with random or non-fixed evaluation approaches, the fixed-value, control-variable sampling strategy enables systematic and reliable assessment of each variable position. Activating one variable at a time using fixed-value subinterval sampling ensures controlled and consistent evaluation. However, evaluating each variable at a single fixed point may overlook its behavior across the full feasible interval, potentially missing non-linear effects or extreme contributions. Therefore, by combining the stratified fixed-value evaluation strategy with the interval-based uniform sampling evaluation strategy, a more comprehensive and meaningful assessment of variable importance can be achieved, thereby balancing stability and coverage. The pseudo-code is presented in Algorithm 1.

3.2. Design of action space enhancement and dynamic fusion mechanisms

State evaluation in the reinforcement learning module plays a critical role in enhancing both decision-making efficiency and exploration effectiveness for large-scale sparse multi-objective optimization. During

Table 3
The mean and standard deviation of IGD results obtained by SMAF and its comparison algorithms on the SMOP1-SMOP8 three-objective benchmark problems, where the best result in each row is highlighted. ‘+’, ‘-’, and ‘≈’ indicate that the result by another MOEA is better than, worse than, or comparable to that obtained by SMAF.

Problem	D	MSKEA	S-NSGA-II	SparseEA	SparseEA2	AGSEA	S-ECSO	MOEA/PSL	SMAF
SMOP1	500	7.0960e-2 (1.94e-3) -	6.1136e-2 (4.32e-3) -	8.9491e-2 (3.60e-3) -	8.6523e-2 (3.20e-3) -	4.6389e-2 (1.36e-3) -	7.5615e-2 (1.57e-3) -	1.0023e-1 (7.61e-3) -	**4.3185e-2 (7.83e-4)**
	1000	7.1153e-2 (1.58e-3) -	6.1671e-2 (5.29e-3) -	8.6638e-2 (3.41e-3) -	8.8153e-2 (8.84e-3) -	4.4960e-2 (1.11e-3) -	7.5494e-2 (9.80e-4) -	9.7426e-2 (3.29e-3) -	**4.2502e-2 (4.26e-4)**
	2000	7.1017e-2 (1.27e-3) -	6.4784e-2 (3.25e-3) -	8.5546e-2 (4.34e-3) -	9.0766e-2 (1.45e-2) -	4.4695e-2 (1.45e-3) -	7.5305e-2 (7.15e-4) -	9.4762e-2 (3.26e-3) -	**4.2457e-2 (4.61e-4)**
SMOP2	500	1.2229e-1 (4.29e-3) -	7.5211e-2 (6.04e-3) -	1.4663e-1 (5.43e-3) -	1.4372e-1 (4.34e-3) -	6.0431e-2 (3.21e-3) -	1.3663e-1 (3.78e-3) -	1.6326e-1 (5.00e-3) -	**5.4496e-2 (1.48e-3)**
	1000	1.2266e-1 (4.54e-3) -	8.1575e-2 (7.43e-3) -	1.4491e-1 (6.40e-3) -	1.4458e-1 (3.62e-3) -	5.9310e-2 (2.78e-3) -	1.3530e-1 (1.71e-3) -	1.6109e-1 (3.23e-3) -	**5.3650e-2 (1.81e-3)**
	2000	1.2392e-1 (2.43e-3) -	9.3810e-2 (7.17e-3) -	1.4108e-1 (4.05e-3) -	1.4469e-1 (3.50e-3) -	5.9691e-2 (2.66e-3) -	1.3573e-1 (1.84e-3) -	1.6210e-1 (4.20e-3) -	**5.4365e-2 (1.75e-3)**
SMOP3	500	6.4344e-2 (5.49e-3) -	6.0336e-2 (4.71e-3) -	9.3724e-2 (3.80e-3) -	8.8770e-2 (2.92e-3) -	4.3663e-2 (7.94e-4) -	2.9280e-1 (1.77e-1) -	1.6301e-1 (4.10e-1) -	**4.2889e-2 (6.91e-4)**
	1000	6.7176e-2 (4.73e-3) -	6.1937e-2 (2.98e-3) -	9.5585e-2 (3.50e-3) -	9.3852e-2 (3.05e-3) -	4.5348e-2 (9.80e-4) -	1.2814e-1 (5.52e-2) -	1.9006e-1 (4.06e-1) -	**4.3775e-2 (7.98e-4)**
	2000	6.6006e-2 (4.02e-3) -	6.5031e-2 (3.29e-3) -	9.5638e-2 (3.72e-3) -	9.5945e-2 (1.98e-3) -	4.7392e-2 (1.68e-3) -	1.0013e-1 (1.69e-2) -	3.4757e-1 (6.79e-1) -	**4.4024e-2 (6.24e-4)**
SMOP4	500	3.0123e-2 (5.44e-4) +	7.2845e-2 (1.43e-2) -	6.2494e-2 (8.64e-3) -	5.6423e-2 (4.44e-3) -	3.0343e-2 (3.88e-4) +	**2.9120e-2 (6.07e-4)** +	6.2626e-2 (1.71e-2) -	3.0690e-2 (6.34e-4)
	1000	3.0326e-2 (5.39e-4) =	7.2450e-2 (1.45e-2) -	6.4200e-2 (1.03e-2) -	6.0723e-2 (1.04e-2) -	3.0560e-2 (6.37e-4) =	**2.9176e-2 (6.03e-4)** +	5.8884e-2 (1.34e-2) -	3.0576e-2 (6.13e-4)
	2000	3.0178e-2 (5.37e-4) =	7.1933e-2 (1.10e-2) -	6.2982e-2 (8.72e-3) -	5.8583e-2 (7.56e-3) -	3.0379e-2 (5.63e-4) =	**2.8938e-2 (5.05e-4)** +	6.3831e-2 (1.14e-2) -	3.0392e-2 (6.92e-4)
SMOP5	500	3.7585e-2 (1.84e-3) -	7.0173e-2 (1.03e-2) -	7.0836e-2 (1.20e-2) -	6.4462e-2 (1.24e-2) -	3.4682e-2 (8.53e-4) -	3.7740e-2 (9.48e-4) -	6.0484e-2 (8.24e-3) -	**3.1921e-2 (8.41e-4)**
	1000	3.7252e-2 (1.90e-3) -	7.6403e-2 (1.50e-2) -	7.0201e-2 (1.06e-2) -	6.7360e-2 (1.12e-2) -	3.2636e-2 (1.04e-3) =	3.9262e-2 (1.20e-3) -	6.5110e-2 (8.24e-3) -	**3.2224e-2 (7.77e-4)**
	2000	3.7108e-2 (1.17e-3) -	7.9774e-2 (1.15e-2) -	7.0574e-2 (1.08e-2) -	6.9048e-2 (1.43e-2) -	3.1658e-2 (5.69e-4) -	3.8909e-2 (7.79e-4) -	6.5254e-2 (1.08e-2) -	**3.1534e-2 (7.45e-4)**
SMOP6	500	3.7419e-2 (2.73e-3) -	6.8180e-2 (7.87e-3) -	7.4515e-2 (9.11e-3) -	6.3886e-2 (5.28e-3) -	3.4011e-2 (9.42e-4) -	4.0660e-2 (1.70e-3) -	6.7624e-2 (1.21e-2) -	**3.2418e-2 (9.57e-4)**
	1000	3.7514e-2 (3.24e-3) -	7.8734e-2 (1.26e-2) -	7.3777e-2 (7.61e-3) -	6.4784e-2 (8.28e-3) -	3.2713e-2 (7.56e-4) -	4.2046e-2 (2.47e-3) -	7.8306e-2 (2.41e-2) -	**3.1497e-2 (5.89e-4)**
	2000	3.7830e-2 (2.36e-3) -	7.7440e-2 (1.06e-2) -	7.2451e-2 (8.70e-3) -	6.6192e-2 (7.58e-3) -	3.2305e-2 (6.97e-4) -	4.2288e-2 (2.36e-3) -	7.0710e-2 (1.33e-2) -	**3.1295e-2 (6.14e-4)**
SMOP7	500	1.9076e-1 (9.86e-3) -	1.0339e-1 (1.01e-2) -	2.1688e-1 (7.38e-3) -	2.1704e-1 (6.84e-3) -	1.6001e-1 (2.51e-2) -	2.3229e-1 (4.03e-3) -	2.7949e-1 (1.50e-1) -	**6.2950e-2 (1.70e-3)**
	1000	1.9620e-1 (6.33e-3) -	1.1518e-1 (1.16e-2) -	2.1844e-1 (6.61e-3) -	2.2150e-1 (5.95e-3) -	1.7870e-1 (1.79e-2) -	2.3128e-1 (3.13e-3) -	2.6669e-1 (1.02e-1) -	**6.2912e-2 (9.03e-4)**
	2000	1.9922e-1 (3.95e-3) -	1.3178e-1 (1.30e-2) -	2.1346e-1 (6.99e-3) -	2.2481e-1 (3.74e-3) -	1.9777e-1 (1.91e-2) -	2.3166e-1 (4.83e-3) -	2.5089e-1 (6.59e-2) -	**6.3584e-2 (1.31e-3)**
SMOP8	500	3.5611e-1 (2.54e-2) -	**1.9554e-1 (2.04e-2)** +	4.2841e-1 (1.71e-2) -	4.2258e-1 (2.49e-2) -	3.4624e-1 (3.05e-2) -	5.0670e-1 (3.15e-2) -	4.6669e-1 (3.22e-2) -	2.2953e-1 (1.40e-2)
	1000	3.9117e-1 (2.05e-2) -	**2.1030e-1 (1.72e-2)** =	4.4051e-1 (1.43e-2) -	4.4689e-1 (1.36e-2) -	3.7978e-1 (2.68e-2) -	5.0564e-1 (1.34e-2) -	5.2523e-1 (1.02e-2) -	2.1573e-1 (1.14e-2)
	2000	4.1165e-1 (8.32e-3) -	2.2885e-1 (1.68e-2) -	4.5237e-1 (1.01e-2) -	4.7277e-1 (9.84e-3) -	4.2907e-1 (3.21e-2) -	4.9447e-1 (1.02e-2) -	5.3621e-1 (4.13e-3) -	**2.1208e-1 (9.47e-3)**
+/-/=		1/21/2	1/22/1	0/24/0	0/24/0	1/20/3	3/21/0	0/24/0	

Table 4
The mean values and standard deviations of HV results obtained by SMAF and its competitors on the practical test problems. The best result in each row is highlighted by black bold.

Problem	D	S-ECSO	S-NSGA-II	SMAF
SR1	1024	1.9220e-1 (7.18e-3) -	1.1428e-1 (5.28e-3) -	**3.9470e-1 (1.75e-3)**
SR2	2048	1.9123e-1 (8.07e-3) -	1.0207e-1 (3.22e-3) -	**3.7882e-1 (1.37e-3)**
SR3	5120	1.7507e-1 (6.12e-3) -	9.5257e-2 (1.35e-3) -	**3.7088e-1 (1.39e-3)**
+/-/=		0/3/0	0/3/0	

Table 5
The mean values and standard deviations of HV results obtained by SMAF and its competitors on the practical test problems. The best result in each row is highlighted by black bold.

Problem	D	AGSEA	MSKEA	SparseEA	SparseEA2	MOEAPSL	SMAF
SR1	1024	3.8804e-1 (2.92e-3) -	3.4182e-1 (6.78e-3) -	3.8957e-1 (3.25e-3) -	3.8914e-1 (4.46e-3) -	3.0790e-1 (1.69e-2) -	**3.9470e-1 (1.75e-3)**
SR2	2048	3.6878e-1 (2.66e-3) -	3.0871e-1 (6.74e-3) -	3.7280e-1 (3.86e-3) -	3.7336e-1 (3.99e-3) -	2.8760e-1 (1.16e-2) -	**3.7882e-1 (1.37e-3)**
SR3	5120	3.5887e-1 (1.37e-3) -	2.9182e-1 (4.17e-3) -	3.6922e-1 (2.95e-3) -	3.6908e-1 (2.69e-3) -	2.4159e-1 (5.01e-3) -	**3.7088e-1 (1.39e-3)**
FS1	166	9.8235e-1 (7.46e-3) -	9.6831e-1 (8.83e-3) -	9.8033e-1 (8.28e-3) -	9.6941e-1 (8.75e-3) -	9.8119e-1 (4.54e-3) -	**9.8681e-1 (4.65e-3)**
FS2	256	9.2596e-1 (5.49e-3) -	9.1736e-1 (4.75e-3) -	9.2423e-1 (4.62e-3) -	9.2098e-1 (5.04e-3) -	9.2774e-1 (4.10e-3) -	**9.3206e-1 (5.67e-3)**
FS3	310	9.9826e-1 (2.33e-4) =	9.9228e-1 (1.36e-2) =	9.9585e-1 (9.06e-3) =	9.8492e-1 (1.75e-2) -	9.9811e-1 (1.15e-3) =	**9.9829e-1 (2.50e-4)**
CN1	234	9.7335e-1 (5.16e-4) -	9.7354e-1 (6.35e-4) -	9.7296e-1 (4.29e-4) -	9.7193e-1 (8.25e-4) -	9.6973e-1 (1.17e-2) -	**9.7462e-1 (4.61e-4)**
CN2	311	9.2083e-1 (2.58e-3) -	9.2464e-1 (2.76e-3) -	9.2038e-1 (2.28e-3) -	9.1401e-1 (4.18e-3) -	9.2283e-1 (3.23e-3) -	**9.2945e-1 (1.88e-3)**
CN3	452	9.9621e-1 (1.56e-5) -	9.9620e-1 (1.85e-5) -	9.9621e-1 (9.08e-6) -	9.9610e-1 (6.61e-5) -	9.9435e-1 (4.62e-3) -	**9.9622e-1 (1.51e-6)**
PM1	500	1.9975e-1 (4.49e-3) -	1.9768e-1 (4.46e-3) -	2.0429e-1 (3.18e-3) =	1.7176e-1 (3.95e-3) -	1.6294e-1 (6.39e-2) -	**2.0574e-1 (3.66e-3)**
PM2	1000	1.9108e-1 (3.99e-3) -	1.9194e-1 (4.57e-3) -	2.0094e-1 (4.55e-3) =	1.7082e-1 (3.03e-3) -	1.3621e-1 (8.41e-2) -	**2.0246e-1 (4.48e-3)**
PM3	2000	1.7509e-1 (2.50e-3) -	1.8024e-1 (4.37e-3) -	1.8980e-1 (5.26e-3) =	1.5814e-1 (3.68e-3) -	1.2895e-1 (8.07e-2) -	**1.9164e-1 (3.91e-3)**
+/-/=		0/11/1	0/11/1	0/8/4	0/11/0	0/11/1	

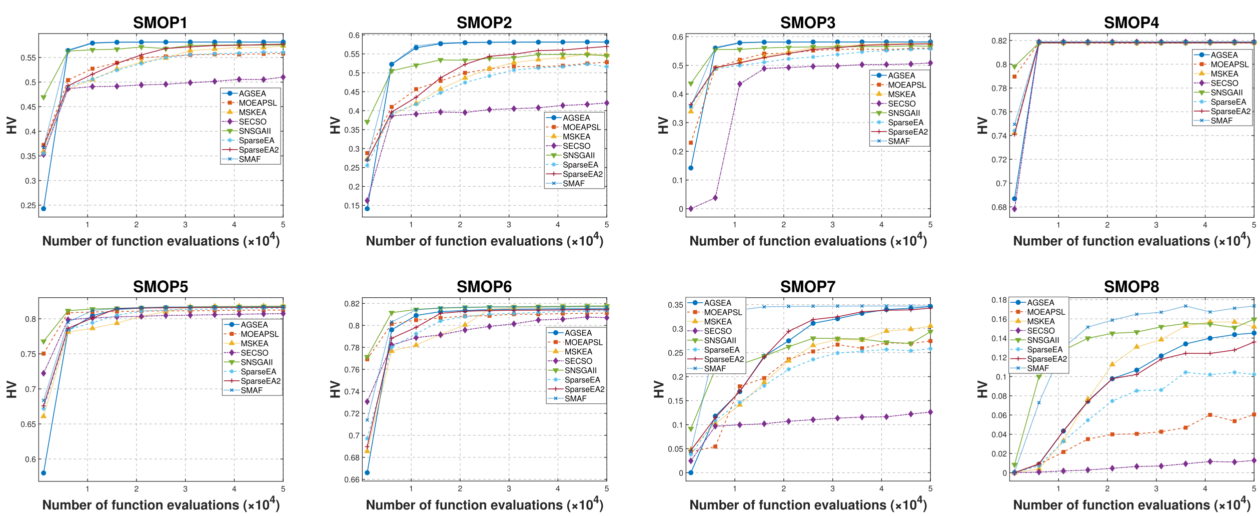

Fig. 5. The HV values (median) obtained by SMAF and its comparison algorithms on the bi-objective SMOP1-SMOP8, with 500 decision variables under the number maximum of fitness evaluations varying from 1000 to 50,000.

action design, accurately evaluating the potential of current solutions guides the policy network toward more promising search directions.

In the AGSEA algorithm, action design is closely integrated with fitness evaluation mechanisms, which can be broadly categorized into three types: (1) directly using the existing fitness values of individuals; (2) computing fitness based on variable mask information to reflect variable selection states; and (3) generating fitness via stochastic perturbation to promote search diversity.

In high-dimensional and non-uniform decision spaces, Single-variable perturbation evaluation based on Latin hypercube uniform sampling often suffers from interference among multiple variables when evaluating individual contributions, making it difficult to reliably reveal complex variable-objective relationships. To address this limitation, we introduce an additional structured evaluation action based on fixed-value control of individual variables. This action systematically identifies the internal structure of the variable space, obtains reliable prior importance information, and guides the generation of subsequent sparse mask structures, thereby enhancing sensitivity to key variables and improving search efficiency.

Furthermore, stochastic perturbation-based evaluation can fluctuate due to its randomness and limited sensitivity to structural characteristics. To improve stability and adaptability, a dynamic weighted fusion strategy is introduced to adaptively combine complementary sources of fitness evaluation. Specifically, the algorithm considers three complementary fitness evaluation actions, as detailed below:

- F_1: the fitness estimated via uniform sampling;
- F_2: the fitness from fixed-value variable evaluation;
- ϵ: a random noise vector to maintain exploration.

Algorithm 1: Hierarchical Structured Evaluation Strategy.

Input: N (population size), D (number of decision variables),K (number of intervals)
Output: *Dec* (decision matrix), $Mask$ (mask matrix)
`// Group variables by bounds`
Group variables by bounds into *result*;
$index$ ← the set of unique group IDs in *result*;
Divide decision space into three equal intervals;
for $j \leftarrow 1$ **to** K **do**
  for $i \leftarrow 1$ **to** *length(index)* **do**
    `// Calculate lower and upper bounds of the j-th subinterval`
    $l1$ ← lower bound of the j-th subinterval of variable $index(i)$;
    $u1$ ← upper bound of the j-th subinterval of variable $index(i)$;
    $m1$ ← midpoint between $l1$ and $u1$;
    `// Prevent degenerate or invalid values`
    if $m1 == 0$ **then**
      $m1 \leftarrow u1$;
    Dec ← Generate the corresponding decision variable matrix;
    $Mask$ ← Generate an identity matrix;
    Evaluate the masked decision variables;
    $f1$ ← Fitness cumulative ranking value;
return *Dec*, $Mask, f1$

Here, F_1 and F_2 in the framework diagram (Fig. 1) are consistent with the definitions provided above.

Among these, the third action relies solely on random perturbations, which, while useful for maintaining diversity, may introduce excessive fluctuations and insufficient responsiveness to the structural characteristics of the decision variable space. To address this limitation, a dynamic weighted fusion strategy is introduced, which adaptively integrates the different sources of fitness information. The fused fitness is defined as follows (see Eq. (3)):

$$F = (1 - \beta)F_1 + \beta F_2 + \alpha\varepsilon, \tag{3}$$

where $F \in \mathbb{R}^D$ is the final fitness vector used for action evaluation, D is the number of decision variables, F_1 represents the guidance vector obtained from decision variable evaluation using uniform sampling, emphasizing exploration; F_2 represents the guidance vector obtained from decision variable evaluation using fixed-value sampling, which focuses on a systematic assessment of each decision variable position and ε is random noise to preserve controlled stochasticity. The adaptive coefficients β and α adjust the relative contributions of these components over time, guiding the search from global exploration toward local exploitation.

The coefficient $\beta \in [0, 1]$ controls the balance between uniform and structured fitness components. To gradually shift the evaluation emphasis from exploration to exploitation, β is designed to increase with the number of fitness evaluations (see Eq. (4)):

$$\beta = 1 - \exp\left(-\kappa \cdot \frac{\mathrm{FE}}{\mathrm{maxFE}}\right), \tag{4}$$

where FE is the current number of fitness evaluations, maxFE is the maximum allowed, and $\kappa > 0$ is a scaling parameter (empirically set to 5) that controls the transition speed.

Meanwhile, $\alpha \in [0, 1]$ determines the magnitude of random perturbations. To encourage exploration in the early stage and gradually reduce stochasticity for convergence, α decays exponentially with iterations (see Eq. (5)):

$$\alpha = \exp\left(-\kappa \cdot \frac{\mathrm{FE}}{\mathrm{maxFE}}\right). \tag{5}$$

Initially, the search is dominated by random perturbations (α high), promoting exploration (see Fig. 2). As the number of fitness evaluations increases, β gradually increases, emphasizing structured evaluation, while α decays, reducing randomness. The crossover point indicates a balance between exploration and exploitation.

While the fusion strategy improves fitness evaluation, its effectiveness also depends on the action selection policy that determines which candidate moves are executed. By combining the adaptive weights β and α, this fusion strategy enables a smooth transition from uniform to structured fitness evaluation. At the same time, controlled randomness is maintained to enhance exploration and convergence stability throughout the search process.

In the original AGSEA, action selection follows a greedy strategy based on the outputs of a reinforcement learning network. The network assigns a preference score to each candidate action, and the action with the highest score is selected deterministically Li et al. (2024). This simple strategy can achieve rapid convergence in low-dimensional problems or scenarios where the network is well-trained. However, in large-scale sparse multi-objective optimization problems, where the search space is high-dimensional and feedback signals are sparse or noisy, the deterministic greedy selection may reduce exploration diversity, potentially limiting the algorithm's ability to thoroughly explore the solution space during early learning stages.

Algorithm 2: UsingNet: Softmax-Based Action Selection.

Input: Net (neural net), Memory (experience mdemory), Fitness1, Fitness2, Fitness3, M (mask), Action, ANum (action num)
Output: Act (selected action), Fitness
if *in early stage or Memory insufficient* **then**
  Select a random action;
else
  `CurrentState` ← last entry in `Memory`;
  for $i = 1$ **to** *ANum* **do**
    `// Combine state and action into a single input vector`
    `input` ← concatenate(`CurrentState`, i)
    `output[i]` ← `Net(input)`;
  `// Apply softmax transformation`
  `output` ← `output` - max(`output`);
  `prob` ← exp(`output`)/ $\sum$ exp(`output`);
  if *probabilities are invalid* **then**
    Set `prob` as uniform distribution;
  Sample `Action` from the probability distribution `prob`;
switch `Action` **do**
  case *1* **do**
    `Fitness` ← `Fitness1`;
  case *2* **do**
    `Fitness` ← number of zero entries in `M`;
    if *only one individual* **then**
      `Fitness` ← `M`
  case *3* **do**
    $\beta \leftarrow 1 - \exp(-5 \cdot$ `FE/maxFE`$)$;
    $\alpha \leftarrow \exp(-5 \cdot$ `FE/maxFE`$)$;
    `Fitness` ← $(1 - \beta) \cdot$ `Fitness1` $+ \beta \cdot$ `Fitness2`;
    `Fitness` ← `Fitness` + $\alpha \cdot$ uniform noise;
  case *4* **do**
    `Fitness` ← `Fitness2`;
return *Act*, *Fitness*

To address the limitations of the original greedy action selection strategy, which deterministically chooses the action with the highest network preference and may lead to premature convergence and insufficient exploration, we adopt a Softmax-based action selection mechanism. This approach transforms the raw preference values produced by the neural network into a probability distribution, allowing stochastic sampling of actions and thereby balancing exploration and exploitation. Specifically, the Softmax function maps a set of scores $\{z_i\}$ into probabilities as (see Eq. (6)):

$$p_i = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)}, \tag{6}$$

where τ is a temperature parameter controlling the randomness of selection. A lower τ sharpens the distribution, favoring exploitation, while a higher τ smooths it, encouraging exploration.

Formally, for each action a_i, the selection probability is defined as follows (see Eq. (7)):

$$P(a_i) = \frac{e^{z_i}}{\sum_{j=1}^{|\mathcal{A}|} e^{z_j}}, \quad z_i = \text{sim}(\text{net}, [\text{state}, a_i]) \tag{7}$$

Here, z_i denotes the preference score output by the neural network when given the concatenation of the current state and action a_i as input. The exponential function non-linearly amplifies the differences among preference values, enhancing the contrast between higher and lower scoring actions. The normalization ensures a valid probability distribution over all candidate actions. Consequently, actions with higher preference values have a higher chance of being selected, promoting exploitation of promising strategies, while lower-valued actions retain a non-zero selection probability, preserving exploration and mitigating the risk of getting trapped in local optima. The pseudo-code is shown in Algorithm 2.

3.3. An auxiliary mask adjustment mechanism guided by sparsity interval feedback

In sparse multi-objective optimization, the sparsity level of masks directly affects the effectiveness of variable selection and the diversity of solutions Ding et al. (2024). Overly dense masks lead to an expanded search space, reduced search efficiency, and a higher risk of local entrapment; in contrast, overly sparse masks may overlook critical variables, limiting the expressiveness of solutions and degrading optimization quality Ding et al. (2022). Therefore, achieving effective search while maintaining sparse structures has become a key challenge in sparse optimization.

To address this issue, this paper proposes an auxiliary control mechanism based on the sparsity distribution of non-dominated front elites. By dynamically guiding the mask sparsity toward the typical range observed in high-quality solutions, the mechanism improves the directionality and stability of mask evolution, thereby enhancing the algorithm's search performance and convergence capability while preserving structural simplicity. The pseudo-code is shown in Algorithm 3.

In this work, the sparsity S of a mask is formally defined as the proportion of active features (see Eq. (8)):

$$S = \frac{\sum_{i=1}^{D} m_i}{D}, \tag{8}$$

where $m_i \in \{0, 1\}$ indicates whether the i-th feature is active, and D denotes the total number of decision variables. By aggregating the sparsity values of all non-dominated individuals, we obtain the current optimal sparsity interval $[S_{\min}, S_{\max}]$.

During offspring generation, if the sparsity of an offspring's mask falls outside this interval, an adaptive sparsity adjustment mechanism is triggered. This mechanism adjusts the mask by turning certain bits "on" or "off" to gradually align its sparsity with the reference interval. The target sparsity S_{target} is determined as follows (see Eq. (9)):

$$S_{\text{target}} = S_{\min} + r_1(S_{\max} - S_{\min}), \tag{9}$$

Algorithm 3: Sparsity-Guided Mask Adjustment Strategy.

Input: `OffMask`(Offspring mask matrix), `Fitness`(fitness vector), problem dimension D, `Population`
Output: $OffMask$
`// Estimate sparsity range from non-dominated individuals`
`// Extract decision matrix`
`Decs`$_{nd}$ ← `Population.decs` where `FrontNo` = 1;
`// Identify activated variables`
`nonMask` ← `Decs`$_{nd} \neq 0$;
`// Compute sparsity ratios`
`nonS` ← `rowSum(nonMask)`$/D$;
`// Set sparsity bounds`
$S_{\min}$ ← min(`nonS`), $S_{\max}$ ← max(`nonS`);
foreach *offspring individual* i **do**
  Compute current sparsity: $s_i \leftarrow \sum$(`OffMask`$(i,:)$)$/D$;
  if $s_i < S_{\min}$ ***or*** $s_i > S_{\max}$ **then**
    Sample target sparsity s_i^* within $[S_{\min}, S_{\max}]$, optionally extended with random noise;
    if $s_i > s_i^*$ **then**
      Identify active variables: `activeIdx` ← $\{j \mid$ `OffMask`$(i,j) = 1\}$;
      Use fitness to select top or noise-perturbed variables for deactivation;
      Set `OffMask`$(i,$ `selected`$)$ ← 0;
    else
      Identify inactive variables: `inactiveIdx` ← $\{j \mid$ `OffMask`$(i,j) = 0\}$;
      Use fitness to select bottom or noise-perturbed variables for activation;
      Set `OffMask`$(i,$ `selected`$)$ ← 1;
return $OffMask$

where $r_1 \in [0, 1]$ is a random number used to maintain diversity. Additionally, a small probability of perturbation is introduced to expand the search scope (see Eq. (10)):

$$S_{\text{target}} = \begin{cases} \max(S_{\min} - r_2 \cdot 0.1, 0), & \text{with probability } p \\ \min(S_{\max} + r_2 \cdot 0.1, 1), & \text{otherwise,} \end{cases} \tag{10}$$

where $r_2 \in [0, 1]$ is a random variable controlling the magnitude of the perturbation, and $p \in [0, 1]$ is the probability parameter governing the direction of the perturbation. The target sparsity expands downward with probability p and upward with probability $1 - p$. This strategy allows the target sparsity to deviate moderately around the baseline sampled value, thereby enlarging the search space and effectively preventing premature convergence.

To enhance the effectiveness of sparsity control, we propose a hybrid variable selection strategy that integrates deterministic ranking with stochastic soft selection based on Gaussian perturbations Geng et al. (2023). When reducing the number of active features, the algorithm identifies variables with minimal contribution to the current optimization objectives and deactivates them accordingly. This hybrid approach helps preserve essential features while introducing controlled diversity. Conversely, when activating inactive features, the algorithm prioritizes potentially important variables that may have been previously overlooked. The importance scores guiding these decisions are obtained through an indirect analysis of each variable's fitness impact on search performance.

To assess the impact of the proposed sparsity adjustment strategy, ablation experiments were conducted on a set of benchmark sparse multi-objective optimization problems. The results demonstrate improved convergence behavior and enhanced solution diversity. Detailed comparisons are presented in the ablation study section (see Section 5).

Algorithm 4: RL-Guided Evolution.

Input: P_0 (initial population), D (number of decision variables), $G_{\max}$ (max generations), $FE_{\max}$ (max function evaluations)
Output: P_{final} (optimized population)

```
// Randomly initialized or pre-trained weights
Initialize DQN network with weights θ ;
// Stores experience tuples
Initialize replay buffer B ;
for t ← 1 to G_max do
    // Compute population-level state
    for each individual x_i ∈ P_t do
        ρ_i ← as defined in Eq. (11);
    s_t ← [mean(ρ_i), std(ρ_i), FE/FE_max] ;
    // Softmax action selection
    a_t ← SelectAction(s_t, θ) ;
    g_{a_t} ← Guiding vector
    // Crossover & mutation
    O_t ← Offspring population;
    P_{t+1} ← Environmental-Selection(P_t ∪ O_t)
    // Compute reward
    r_t ← Eq. (14);
    // Store experience
    Store (s_t, a_t, r_t, s_{t+1}) in B ;
    // Periodic DQN training
    if Training condition met then
        TrainDQN(B, θ)
    // Update total function evaluations
    FE ← FE + |O_t|
return P_final ← P_{G_max}
```

3.4. Reinforcement learning module

During each generation, the RL module in SMAF interacts with the evolutionary process in a closed feedback loop. The pseudo-code is shown in Algorithm 4. It first perceives the population state by computing the sparsity ρ_i of each individual (i.e., the proportion of nonzero decision variables) and then calculating the population mean $\bar{\rho}$ and standard deviation σ_ρ, along with the fraction of consumed function evaluations $\lambda = FE/FE$_max. These three metrics form the state vector s_t, which is used as input to the DQN (Deep Q-Network) Mnih et al. (2013).

Based on this state, the module selects a guiding vector g_{a_t} via an adaptive ϵ-greedy exploration-exploitation strategy, where early in the search the algorithm favors exploration and later stages favor exploitation. The chosen guiding vector is then used to partition decision variables into critical and non-critical subsets, which determines how crossover and mutation are applied to generate offspring O_t.

The offspring and parent populations are merged and subjected to environmental selection, after which the improvement in multi-objective performance is evaluated using the hypervolume-based reward and the experience is stored in the replay buffer for training. Network updates are delayed until sufficient experience is collected (25 % of the evaluation budget) and performed periodically in later stages using recent transitions and gradient descent steps. This mechanism allows the RL agent to continuously refine its policy and adaptively guide the evolutionary search throughout the optimization process.

This adaptive decision-making mechanism is built upon the DQN framework originally proposed in AGSEA Shao et al. (2024), which is directly inherited in SMAF to ensure consistency and reproducibility while seamlessly integrating into the optimization process. The original AGSEA implementation has been open-sourced on the PlatEMO platform Tian et al. (2017), allowing direct access for verification and reproducibility.

In the following, we provide a detailed description of the RL module, including its state representation, action design, reward definition, and training strategy, to clarify how SMAF inherits and implements the DQN-based decision-making mechanism from AGSEA.

State (s_t): At generation t, the state vector encodes the population-level sparsity and the search progress. Following AGSEA, the sparsity ρ_i of each individual x_i is calculated as the ratio of nonzero decision variables(see Eq. (11)).:

$$\rho_i = \frac{1}{D}\sum_{j=1}^{D} \mathbf{1}(x_{ij} \neq 0), \tag{11}$$

where D is the number of decision variables. The population mean $\bar{\rho}$ and standard deviation σ_ρ are computed across all individuals, and the ratio of consumed function evaluations $\lambda = \text{FE}_t/\text{FE}_{\max}$ indicates the current search progress. These three normalized quantities $s_t = [\bar{\rho}, \sigma_\rho, \lambda]$ form the input of the DQN, each naturally within $[0, 1]$. Before training, all features are standardized to zero mean and unit variance based on statistics collected during the warm-up phase to ensure stable learning.

The rationale behind this compact state design is that $\bar{\rho}$ reflects the global activation level of decision variables (exploration intensity), σ_ρ captures structural diversity within the population (search diversity), and λ provides temporal awareness of the optimization stage (convergence progress). Together, they offer an interpretable and evolution-relevant description of the system state, allowing the RL agent to adapt its decision according to both structural sparsity and search dynamics.

Guiding Vectors: The candidate guiding vectors are explicitly defined as a set of predefined search preference vectors, denoted as $\mathcal{F} = \{F_1, F_2\}$. Each vector encodes a distinct type of prior information to guide the evolutionary search.

The first guiding vector F_1 is adopted from the original AGSEA algorithm as a prior knowledge-based guide. It aims to capture the importance of each decision variable during the early stage of optimization. Specifically, a set of d d-dimensional solution vectors is generated using Latin Hypercube Sampling to form a matrix R, which is then combined with the identity matrix B to construct a test population $Q = R \cdot B$. In this setup, the i-th solution has only its i-th variable nonzero (all others set to zero), thereby isolating the independent effect of each variable. Subsequently, the L_1 norm of the objective vector for each solution is computed and accumulated into the corresponding component of F_1, i.e., $F1_i \leftarrow F1_i + L1_i$. The resulting vector $F_1 = [F1_1, \dots, F1_d]$ reflects the magnitude of the perturbation each variable induces in the objective space – the larger the perturbation, the more influential the variable. As this procedure is performed prior to the actual evolutionary process and does not rely on any fitness feedback or population history, F_1 serves as a sensitivity-based prior guide for the subsequent sparse solution search.

In contrast, the second guiding vector F_2 is proposed in this work as an enhancement to provide structured guidance based on fixed-value evaluations. For each variable i, a test solution is created (see Eq. (12)).

$$\mathbf{x}_i = \mathbf{m}_i \odot \mathbf{d}, \tag{12}$$

where $\mathbf{m}_i$ is a unit vector with the i-th entry equal to one and all others zero, and $\mathbf{d}$ contains the midpoint values of the predefined ranges for all variables. The objective function of each test solution is evaluated, and the corresponding non-dominated rank is accumulated.

This produces a score vector that reflects the relative importance of variables – smaller values indicate better performance in the prior test. Consequently, F_2 provides additional structured prior information for guiding the evolutionary search, complementing the original F_1.

Action (a_t): The action space $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$ consists of four predefined strategies, each associated with a specific guiding vector that reflects a distinct search preference. The first two guiding vectors, F_1 and F_2, correspond to sensitivity-based and structured prior knowledge, respectively, while the remaining actions introduce hybrid and sparsity-focused strategies. Each guiding vector g_{a_t} assigns a relative importance score to each decision variable, indicating its priority in crossover and mutation operations.

Once an action a_t is selected, the corresponding guiding vector g_{a_t} is used to cluster decision variables into "critical" and "non-critical" groups via K-means partitioning. These groups determine which variables participate in crossover and mutation, directly linking the RL decision to the algorithm's variation behavior. Meanwhile, each action also adjusts the fitness evaluation: a_1 uses standard fitness, a_2 emphasizes sparsity, a_3 combines both with a stochastic perturbation, and a_4 focuses purely on sparsity-based evaluation(see Eq. (13)):

$$\begin{aligned} &a_3 : F_3 = (1-\beta)\,F_1 + \beta\,F_2 + \alpha\,\mathrm{rand}(1, D), \\ &a_4 : F_4 = \sum_{j=1}^{D} \mathbf{1}(x_j = 0), \end{aligned} \tag{13}$$

where $\beta = 1 - e^{-5\cdot FE/maxFE}$ balances solution quality and sparsity over time, and $\alpha = e^{-5\cdot FE/maxFE}$ controls the strength of stochastic perturbation that decays across generations.

By mapping the RL agent's discrete action selection to both variable grouping and fitness modulation, the framework adaptively controls sparsity focus. This enables the algorithm to emphasize or relax the activation of specific variables according to the detected population structure and current search stage, achieving a dynamic balance between exploration and exploitation.

Reward (r_t): The reward function follows AGSEA's definition and measures the improvement in the hypervolume (HV) between consecutive generations(see Eq. (14)):

$$r_t = HV_t - HV_{t-1}. \tag{14}$$

This direct performance-based signal reflects the effectiveness of the selected guiding vector in improving the population's Pareto front approximation.

Training strategy: The DQN employs the same lightweight architecture as AGSEA – a fully connected neural network with three hidden layers of 10 neurons each and sigmoid activation functions. The learning rate is set to 10^{-3} and the discount factor to $\gamma = 0.1$. Training is postponed until 25 % of the total function evaluations have been completed, ensuring sufficient transitions are collected in the replay buffer before learning begins. Afterward, the network is updated every 10 generations using the most recent 10 transitions, with each update performing 100 gradient descent iterations. Unlike AGSEA, which uses an ϵ-greedy policy, SMAF adopts an adaptive Softmax-based exploration strategy, where the action probabilities are computed as $p(a_t|s_t) = \mathrm{Softmax}(Q(s_t, a_t; \theta)/\tau_t)$ and the temperature $\tau_t = 1 - \lambda$ gradually decreases as the search proceeds, achieving a smooth transition from exploration to exploitation.

The delayed training start (25 % of the budget) prevents overfitting to unstable early dynamics and ensures a diverse replay buffer. Updating every 10 generations balances computational overhead and adaptation speed, as verified in AGSEA and confirmed in preliminary ablation tests. These choices have been shown to yield stable convergence and effective decision adaptation across different benchmark problems.

The RL module retains the core decision-making principle of AGSEA and embeds it within the broader SMAF framework, forming a interpretable adaptive guiding-vector selection process.

3.5. Integration and synergy of mechanisms in SMAF

The SMAF algorithm integrates its key mechanisms into a continuous and synergistic optimization workflow. Fixed-value control-variable evaluation (F2) generates reliable importance scores for each decision variable, guiding the construction of sparse candidate solutions and providing prior knowledge for subsequent fusion. Uniform sampling fitness (F1) and stochastic noise complement F2 to maintain broad exploration and controlled randomness, preventing the search from prematurely focusing on limited regions. These components are combined through dynamic weighted fusion with adaptive coefficients to produce a fused fitness vector that balances global exploration and local exploitation.

The fused vector informs Softmax-based action selection, which can be viewed as a lightweight reinforcement learning (RL) decision module. This module probabilistically selects promising search directions, i.e., which variables to mutate or update, while retaining exploratory diversity, thereby reducing the risk of local entrapment. The selected actions are further guided by sparsity-aware regulation, where variable activation is adaptively controlled according to the elite sparsity intervals to preserve essential variables and maintain population diversity.

Through this tightly coupled interaction of mechanisms, SMAF effectively leverages the complementary strengths of fixed-value evaluation, dynamic fusion, stochastic action selection, sparsity-guided regulation, and mask adjustment. The RL module acts as a "controller" or policy network, dynamically selecting actions based on the current state and continuously adapting its decisions through feedback from the environment, which ensures effective variable prioritization, balanced exploration and exploitation, stable convergence, and competitive solution quality in large-scale sparse multi-objective optimization.

4. Experimental results and analysis

In this section, a series of systematic experiments were conducted on the PlatEMO Tian et al. (2017) platform to evaluate the performance of the proposed SMAF algorithm and its peer algorithms. The experiments consist of two parts: (1) Eight large-scale multi-objective benchmark problems with sparse characteristics (i.e., the SMOP Tian et al. (2019) test functions) were used to comprehensively evaluate the fundamental performance of the algorithms; (2) SMAF was further applied to four real-world problems scenarios–sparse signal reconstruction problem (SR1-SR3) Liang et al. (2018), critical node detection problem (CN1-CN3) Lalou et al. (2018), feature selection problem (FS1–FS3) Karimi et al. (2023) and pattern mining problem (PM1-PM3) Agrawal and Srikant (1994), to verify its effectiveness and applicability in practical applications. Their sparsity, multi-objective nature, and decision variable mapping are summarized as follows:

- **Sparse Signal Reconstruction** Liang et al. (2018): The goal is to reconstruct a high-dimensional signal $x \in \mathbb{R}^N$ from limited linear observations $b = Ax + \epsilon$, where $A \in \mathbb{R}^{M\times N}$ is the measurement matrix and ϵ denotes noise. Only a small subset of signal coefficients are non-zero ($K \ll N$), yielding a sparse decision space. Sparsity is critical in practice to reduce measurement cost and computational burden. The objectives are: (1) minimizing the proportion of non-zero coefficients (sparsity) and (2) minimizing the reconstruction error $\|Ax - b\|$. The main real-world constraints include the limited number of measurements ($M \ll N$) and the presence of noise. SMAF maps each decision variable directly to a signal coefficient, employing its adaptive mask and evolutionary operators to balance sparsity and reconstruction accuracy.
- **Critical Node Detection** Lalou et al. (2018): This problem aims to identify the most critical nodes in a large network to maximize disruption or enhance robustness. Sparsity arises because only a small fraction of nodes can be selected. Each decision variable is binary, indicating whether a node is chosen. The objectives are: (1) minimizing the proportion of selected nodes and (2) minimizing the network's pairwise connectivity after node removal. Real-world constraints include network connectivity and budget limitations on the number of nodes selected. In SMAF, each variable represents a node, and the algorithm adaptively selects a sparse subset of nodes while evaluating network connectivity to guide the search.
- **Feature Selection** Karimi et al. (2023): The task is to select a subset of features from high-dimensional datasets to improve predictive performance. Sparsity occurs because only a few features are

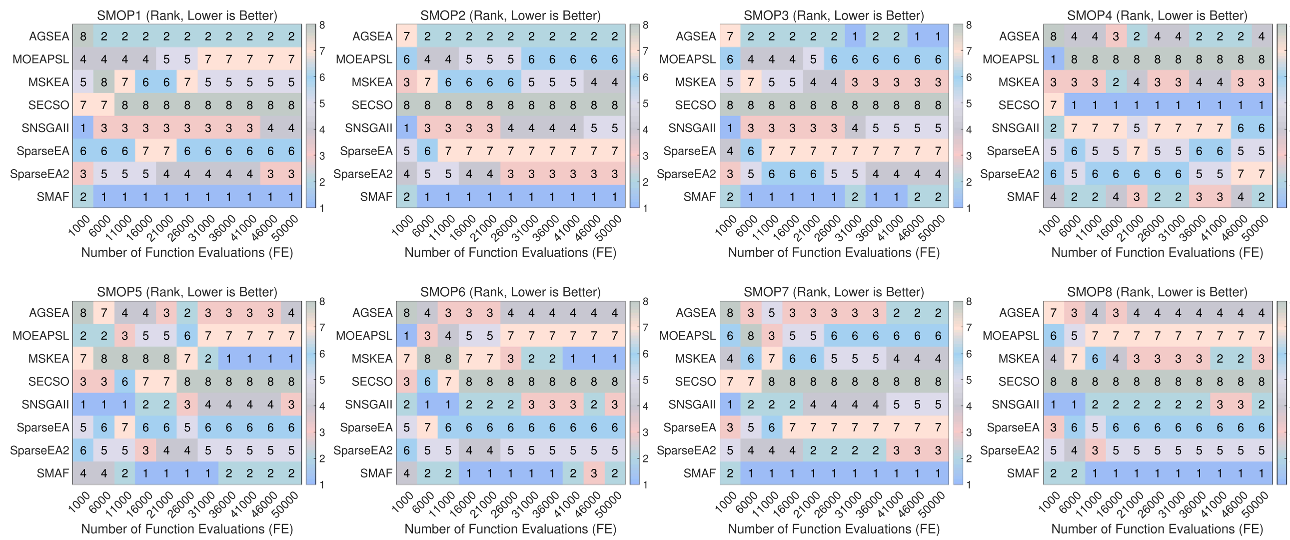

Fig. 6. The ranks of SMAF and its comparison algorithms on the bi-objective SMOP1–SMOP8, calculated based on the median IGD values over 30 independent runs, under different maximum numbers of fitness evaluations ranging from 1000 to 50,000.

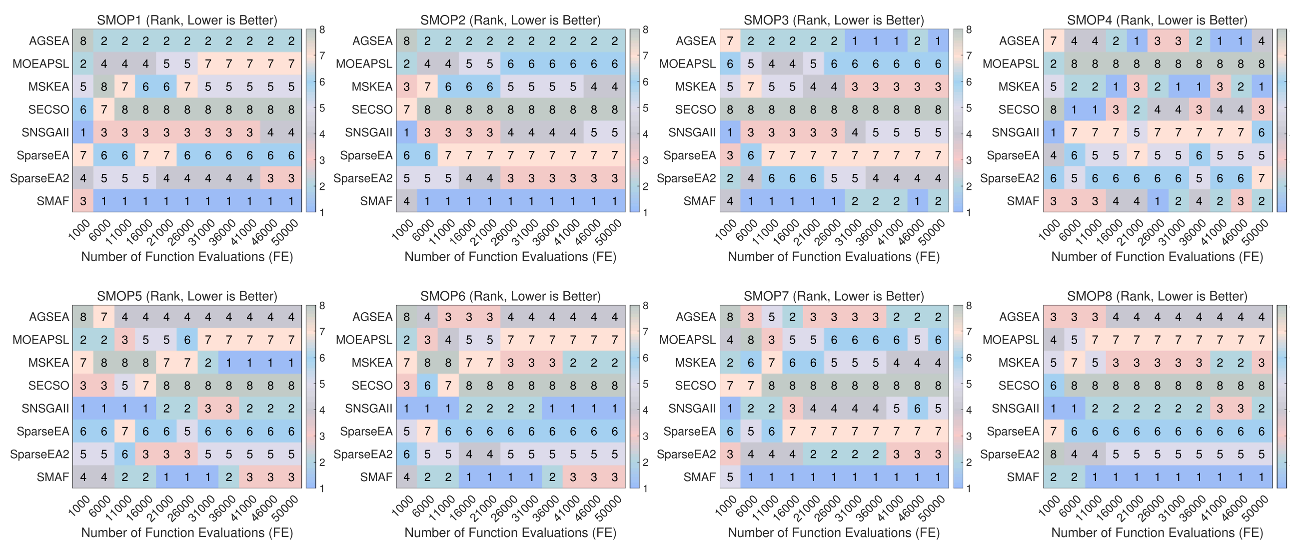

Fig. 7. The ranks of SMAF and its comparison algorithms on the bi-objective SMOP1–SMOP8, calculated based on the median HV values over 30 independent runs, under different maximum numbers of fitness evaluations ranging from 1000 to 50,000.

selected, which helps reduce model complexity and avoid overfitting. Decision variables are binary, indicating whether a feature is included. The objectives are: (1) minimizing the proportion of selected features and (2) minimizing the validation error of the predictive model. Constraints include limited feature budget and dataset-specific dependencies. SMAF applies its mask-guided operators to efficiently explore sparse feature subsets while optimizing predictive accuracy.

- **Pattern Mining** Agrawal and Srikant (1994): The task is to identify a set of frequent patterns from a large transaction dataset. Sparsity occurs because only a small fraction of all possible patterns are meaningful, which helps reduce computational complexity and avoid spurious associations. Decision variables are binary, indicating whether a particular pattern is selected. The objectives are: (1) maximizing the coverage of transactions by the selected patterns and (2) minimizing the redundancy among selected patterns. Constraints include limited number of patterns and pattern-specific dependencies. SMAF applies its mask-guided operators to efficiently explore sparse pattern subsets while optimizing coverage and redundancy.

All experiments were conducted on Ubuntu 6.8.0-55-generic running MATLAB R2025a, on a system with Intel Xeon Silver 4214 CPU (12 cores / 24 threads) and 30 GB RAM with 8 GB swap.

4.1. Settings of algorithms

To ensure fair comparisons, all algorithms adopt a uniform population size of 100 across all test problems. Moreover, each algorithm is executed with a total of 50 × D function evaluations on both the benchmark and real-world problems, where D denotes the number of decision variables. All compared algorithms are implemented using the parameter settings recommended in their original publications.

The compared algorithms include:

- **AGSEA** Shao et al. (2024): Introduces a deep reinforcement learning-based adaptive guide vector selection mechanism, combining variable clustering with sparsity-aware hybrid operators to enhance global exploration and maintain evolutionary directionality.

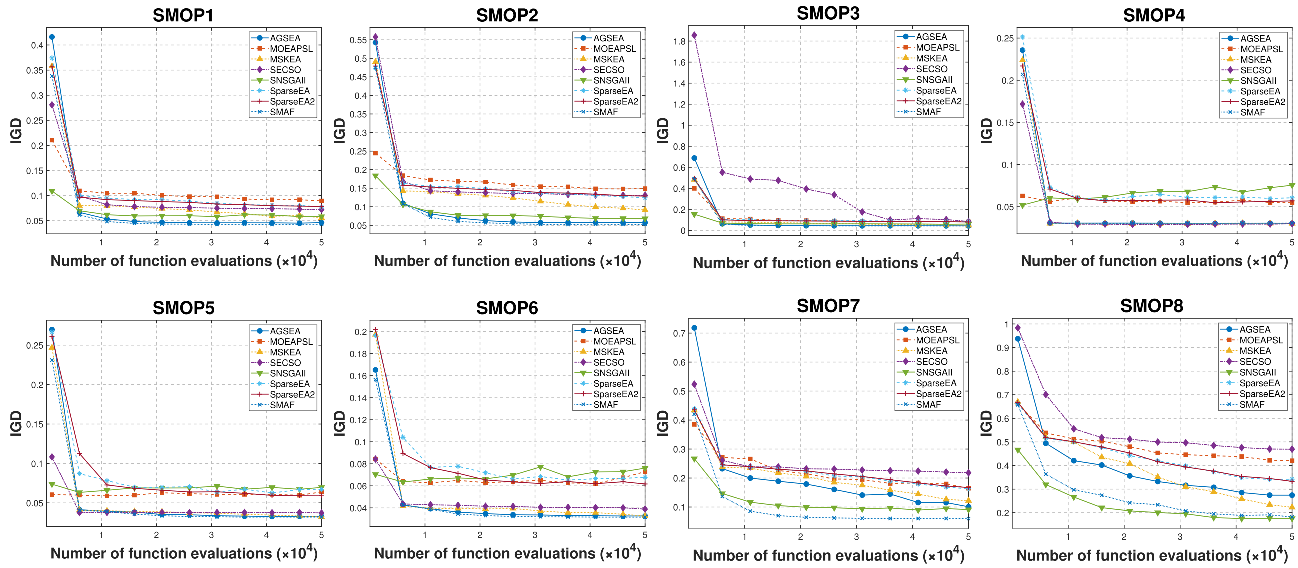

Fig. 8. The IGD values (median) obtained by SMAF and its comparison algorithms on the three-objective SMOP1-SMOP8, with 500 decision variables under the number maximum of fitness evaluations varying from 1000 to 50,000.

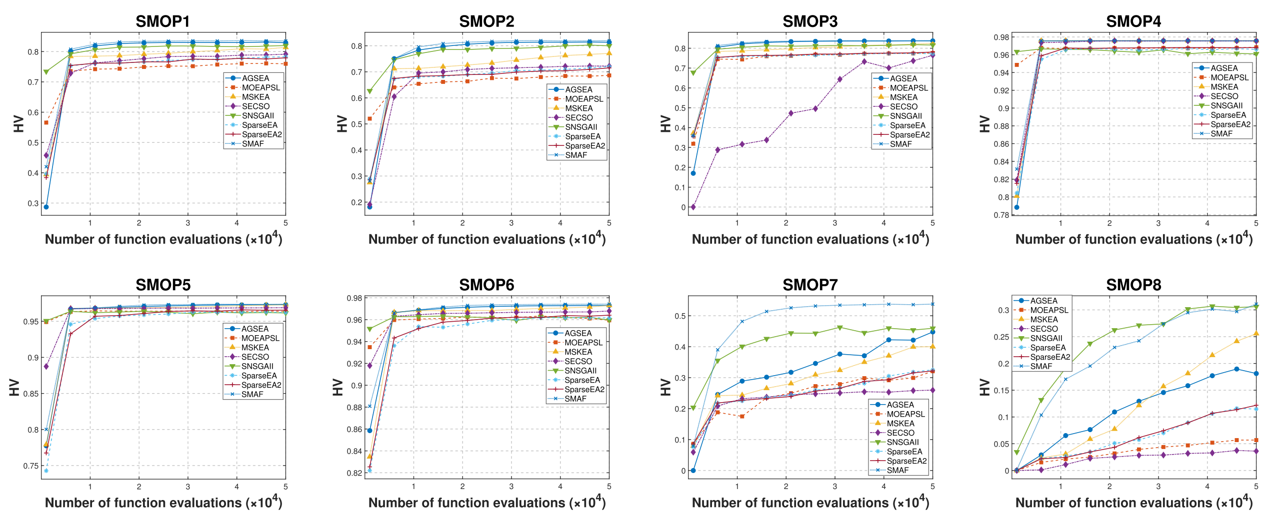

Fig. 9. The HV values (median) obtained by SMAF and its comparison algorithms on the three-objective SMOP1-SMOP8, with 500 decision variables under the number maximum of fitness evaluations varying from 1000 to 50,000.

- **MSKEA** Ding et al. (2022): Employs a multi-stage knowledge-guided evolutionary framework, leveraging historical search knowledge to guide population evolution. By combining reference-point adaptive environmental selection with sparsity-aware mutation, it effectively balances convergence and diversity, improving search efficiency and solution quality in high-dimensional objective spaces.
- **S-NSGA-II** Kropp et al. (2023): Enhances the classical NSGA-II for large-scale sparse multi-objective optimization by introducing variable stripe sparse sampling (VSSPS), sparsity-aware simulated binary crossover (S-SBX), and sparsity-aware polynomial mutation (S-PM). By structuring the initial population and maintaining sparsity in genetic operations, it improves convergence speed and solution sparsity, outperforming traditional NSGA-II and other sparse optimization methods on high-dimensional sparse problems.
- **SparseEA** Tian et al. (2019): Designed for large-scale sparse multi-objective optimization, it employs variable-scoring-based directed initialization and sparsity-preserving genetic operators to guide the search toward sparse Pareto-optimal solutions. Its core innovation lies in using non-dominated sorting to assign importance scores to decision variables, directing population initialization and mutation while maintaining solution sparsity.
- **SparseEA2** Zhang et al. (2023b): Introduces an enhanced variable grouping strategy and adaptive sparsity-preserving mechanism. Compared to traditional large-scale multi-objective optimization algorithms, it more effectively identifies and optimizes key decision variables, showing improved scalability and convergence on high-dimensional sparse problems.
- **S-ECSO** Wang et al. (2021): An enhanced competitive swarm optimization algorithm integrating a triple-competition mechanism and differentiated update strategy to balance exploration and exploitation. By incorporating a strong-convex sparsity operator (SCSparse), it explicitly guides the generation of sparse Pareto-optimal solutions, improving performance on large-scale sparse multi-objective problems.
- **MOEA/PSL** Tian et al. (2020b): Uses RBM Fischer and Igel (2012) and DAE Xue et al. (2012) to learn sparse Pareto-optimal subspaces,

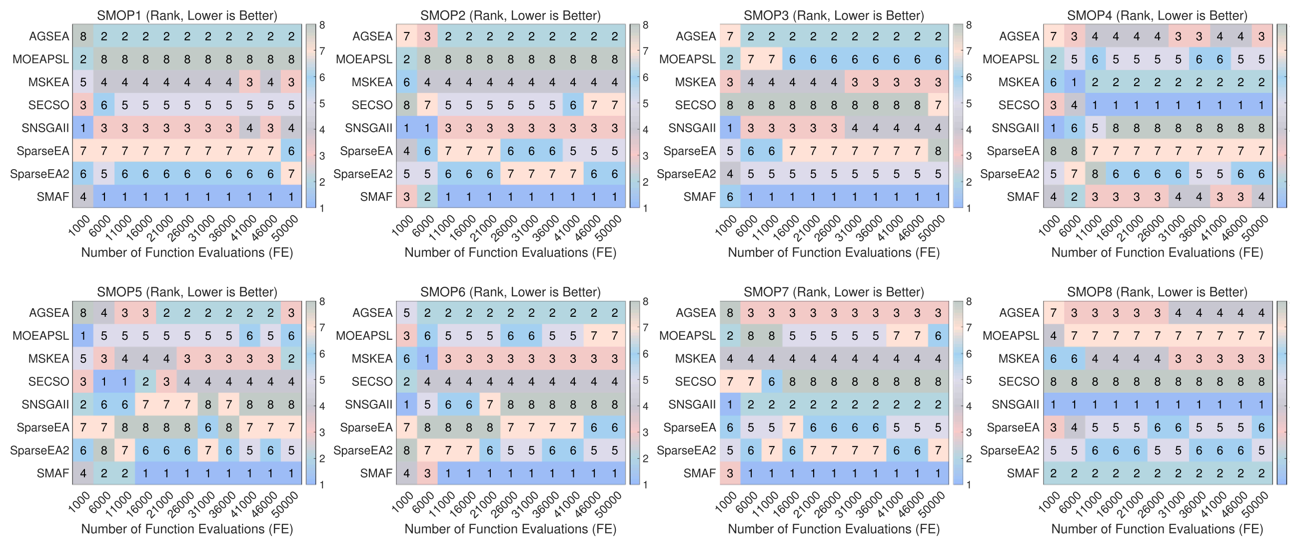

Fig. 10. The ranks of SMAF and its comparison algorithms on the three-objective SMOP1–SMOP8, calculated based on the median IGD values over 30 independent runs, under different maximum numbers of fitness evaluations ranging from 1000 to 50,000.

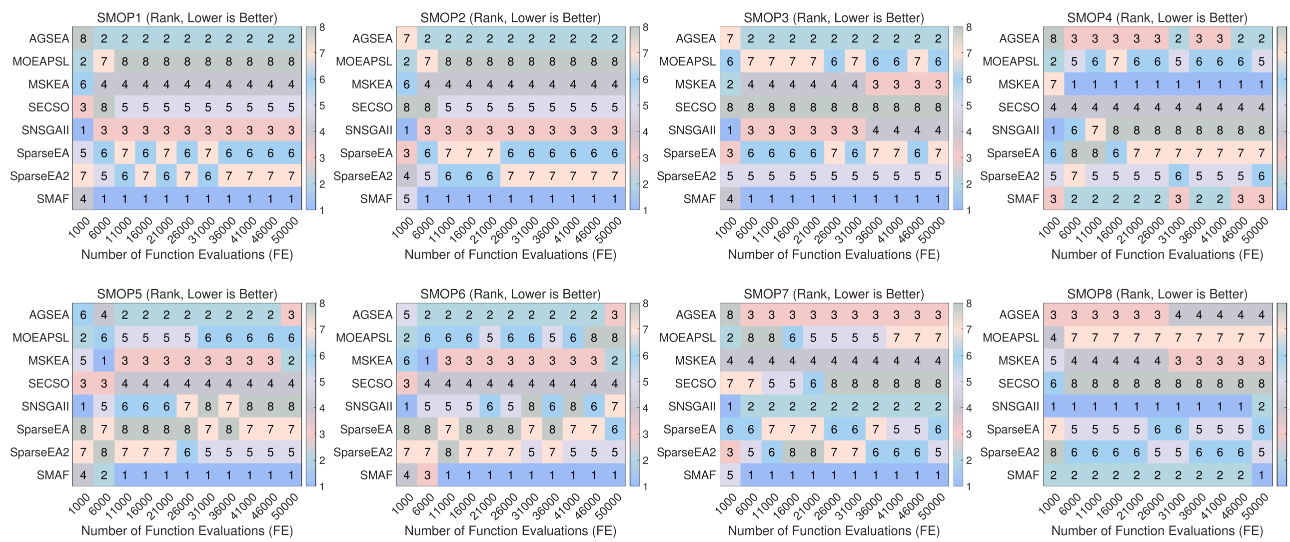

Fig. 11. The ranks of SMAF and its comparison algorithms on the three-objective SMOP1–SMOP8, calculated based on the median HV values over 30 independent runs, under different maximum numbers of fitness evaluations ranging from 1000 to 50,000.

performing genetic operations in the learned low-dimensional space with adaptive tuning, improving efficiency on large-scale sparse multi-objective problems.

4.2. Settings of problems

Table 1 presents the detailed parameter settings for the benchmark problems and real-world applications. For the benchmark problems, the number of decision variables (D) ranges from 500 to 2000, and experiments are conducted with the number of objectives (M) set to 2 and 3. The sparsity level of the optimal solutions is uniformly set to 0.1. The population size (N) is consistently set to 100 for all test problems, ensuring a fair comparison across different problem instances. To comprehensively evaluate the performance of the algorithms on multi-objective optimization problems, two widely used performance metrics are employed in this study: Inverted Generational Distance (IGD) Bosman and Thierens (2003) and Hypervolume (HV) Zitzler and Thiele (1999). In addition, the Wilcoxon rank-sum test Derrac et al. (2011) at a 0.05 significance level is conducted based on 30 independent runs for each test problem to assess the statistical significance of performance differences between the proposed SMAF and the compared algorithms , where the symbols +, -, and ≈ indicate that the result obtained by a compared algorithm is significantly better, significantly worse, and statistically similar to that of the proposed SMAF, respectively.

4.3. Results on benchmark problems

Table 2 presents the statistical results of IGD values obtained by SMAF and the comparison algorithms on the bi-objective benchmark problems SMOP1 to SMOP8. Each result is based on 30 independent runs, with decision variable dimensions ranging from 500 to 2000. The experimental results indicate that SMAF generally performs well on these problems, achieving results comparable to or better than existing methods. According to the experimental results, the SMAF algorithm demonstrates relatively strong performance on the bi-objective benchmark problems SMOP1–SMOP8, achieving the best results on 20 out of 24 test instances. Although it slightly underperforms S-ECSO on SMOP4, the performance gap is minimal. Moreover, SMAF generally yields much lower standard deviations compared to other algorithms, indicating superior stability. Among the comparison algorithms, only S-ECSO shows

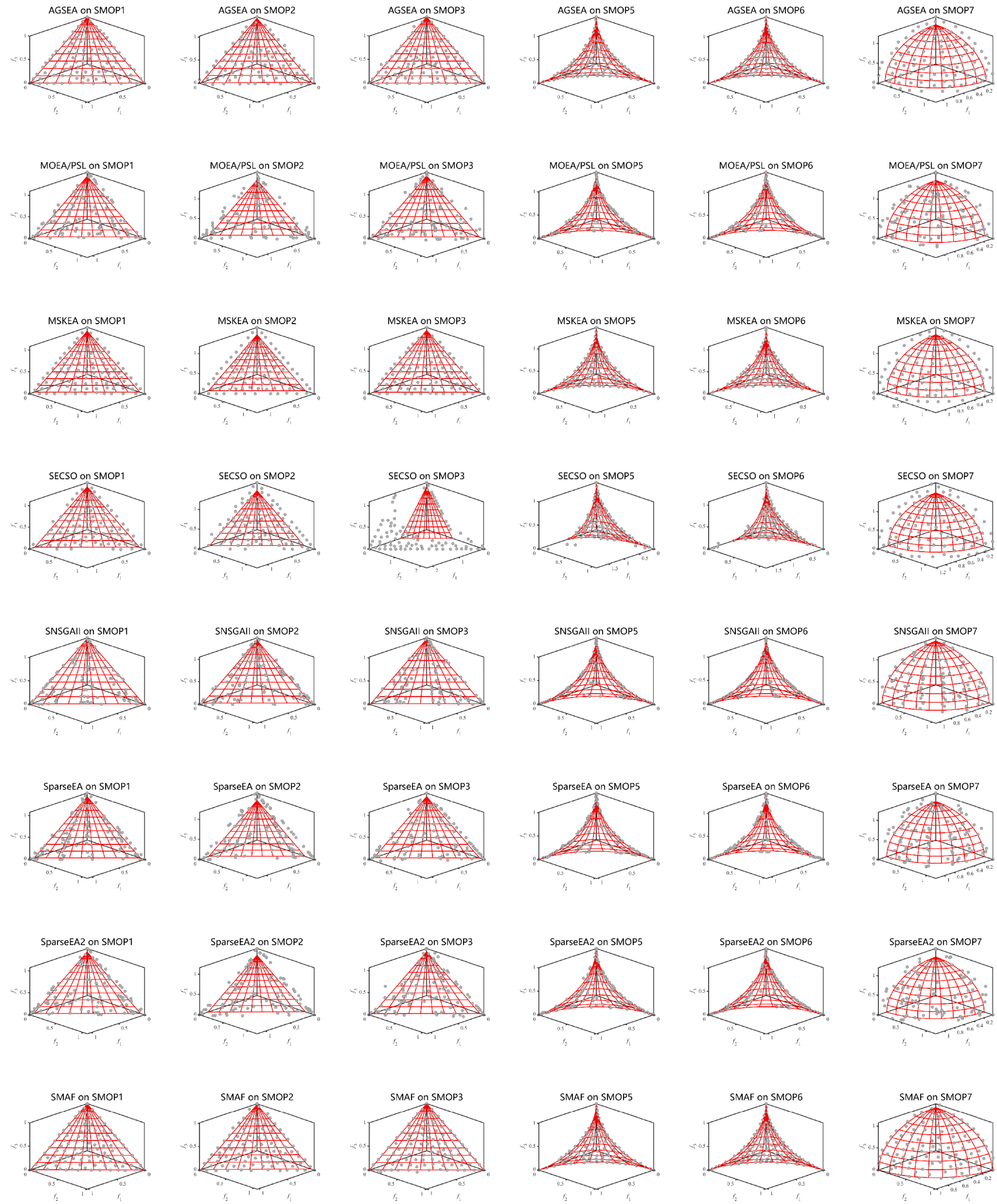

Fig. 12. Populations with the median IGD obtained by SMAF and its competitors on the three-objective SMOP problems with 500 decision variables.

marginally better results on SMOP4 across the three dimensions, while all other algorithms consistently perform no better than, and often worse than, SMAF. In addition, the mean IGD values of SMAF remain remarkably stable as the dimensionality increases from 500 to 2000, demonstrating its robustness and adaptability to problems of varying complexity.

Fig. 3 shows the population distributions of SMAF and the comparison algorithms on the bi-objective SMOP test problems with 500

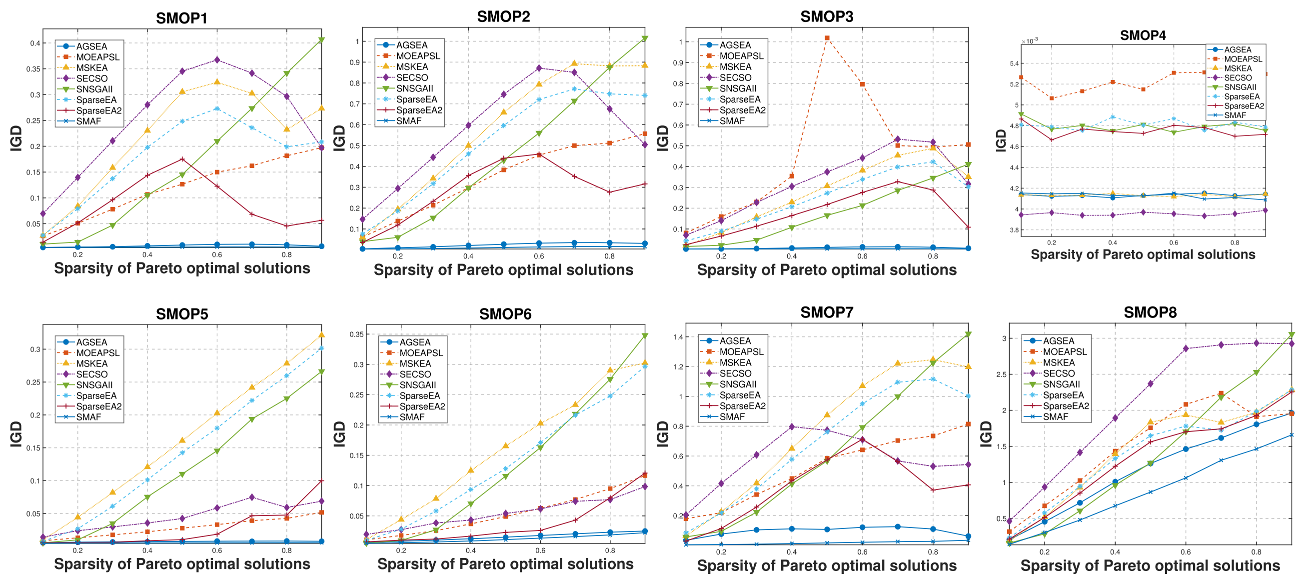

Fig. 13. IGD values obtained by SMAF and its four competitors on the eight SMOP test problems with 500 decision variables and different sparsity.

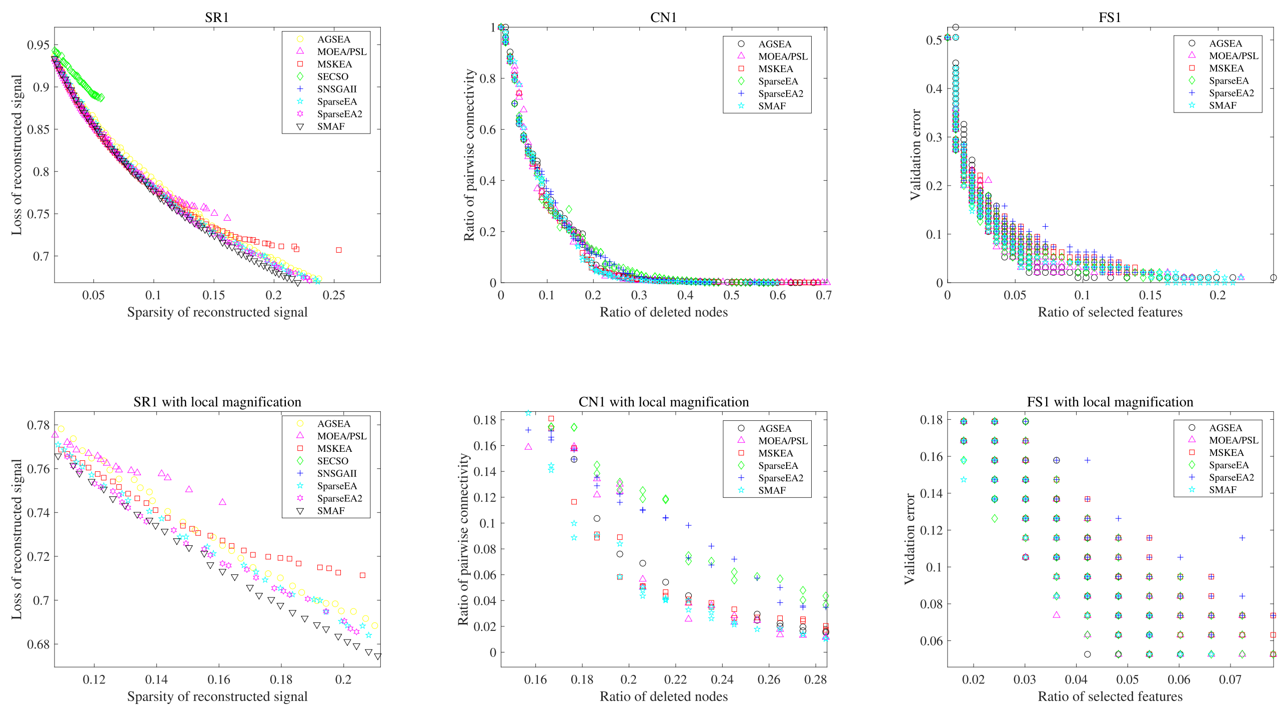

Fig. 14. Populations with the median HV obtained by SMAF and its competitors on SR1, CN1 and FS1.

decision variables. According to the analysis in Table 2, SMAF shows some performance improvements over AGSEA on six of the eight test problems. In particular, it achieves performance gains of 16.97 %, 26.72 %, 87.08 %, and 31.24 % over AGSEA on SMOP2, SMOP6, SMOP7, and SMOP8, respectively.

Figs. 4 and 5 illustrate the trends in the median IGD and HV values obtained by SMAF and the comparison algorithms on the bi-objective benchmark problems SMOP1–SMOP8 with 500 decision variables, as the number of function evaluations increases from 1000 to 5000. It can be observed that SMAF tends to perform well in terms of both convergence and diversity during the early stages of evolution, as suggested by trends in IGD and HV compared with AGSEA. The performance of SMAF is relatively notable on SMOP7 and SMOP8, suggesting its effectiveness and stability in handling complex sparse optimization problems. For the remaining test problems, although SMAF still outperforms AGSEA, the performance gap is relatively small, resulting in nearly overlapping convergence curves in the plots.

Therefore, further analysis should be conducted by considering Figs. 6 and 7 together with the detailed numerical results in the table. These two figures visualize the heatmap rankings of the algorithms based on the Inverted Generational Distance (IGD) and Hypervolume (HV) performance metrics, respectively, evaluated on the bi-objective SMOP1–SMOP8 test problems. The rankings are derived from the median values obtained over 30 independent runs under different maximum fitness evaluation budgets, ranging from 1000 to 50,000. In the ranking scheme, lower numerical ranks correspond to better algorithm performance, where rank 1 indicates the best-performing algorithm, followed by ranks 2, 3, and so forth.

Overall, these observations indicate that SMAF consistently demonstrates competitive convergence and diversity performance across the benchmark problems, highlighting its robustness in handling complex sparse multi-objective optimization tasks.

The experimental results (see Table 3) indicate that the SMAF algorithm generally exhibits competitive performance on the tri-objective benchmark problems SMOP1–SMOP8, particularly in terms of the IGD metric. Among the 24 test instances, SMAF achieved the best results in 19 problems. Although S-ECSO slightly outperforms SMAF on SMOP4, and S-NSGA-II shows marginally better performance on some dimensions of SMOP8, SMAF maintains an advantage on higher-dimensional instances of SMOP8. Moreover, SMAF generally produces smaller standard deviations, suggesting more stable outcomes. Specifically, SMAF performs well across most dimensions (D = 500, 1000, and 2000), with notable improvements on SMOP7 exceeding 60 %, and substantial gains on SMOP8 ranging approximately from 34 % to 51 %. In contrast, SMAF demonstrates comparable performance to AGSEA on SMOP4. In general, SMAF shows promising convergence and stability in most cases, while some instances still leave room for further improvement.

Figs. 8 and 9 illustrate the performance trends in terms of the median IGD and HV values achieved by SMAF and the comparison algorithms on the tri-objective benchmark problems SMOP1–SMOP8, as the number of function evaluations increases from 1000 to 5000. As shown in the figures, SMAF demonstrates consistently superior performance on SMOP1, SMOP2, SMOP6, and SMOP7. It maintains clear advantages over the other algorithms in terms of both IGD and HV throughout the entire optimization process.

To provide a more comprehensive comparison, Figs. 10 and 11 display the corresponding heatmap rankings of the algorithms based on the same median IGD and HV values obtained over 30 independent runs. These heatmaps clearly illustrate the performance distribution and dynamic ranking changes of the algorithms as the number of function evaluations increases. The varying color intensities highlight differences in algorithm effectiveness and stability across different evaluation budgets in the more complex tri-objective scenarios.

On the SMOP8 test problems, although S-NSGA-II maintains a consistent lead throughout the evaluation process, the performance curve of SMAF shows stable convergence and a continuous narrowing of the performance gap. Notably, by the end of the evaluation (around 50,000 evaluations), SMAF nearly matches S-NSGA-II's performance, indicating strong long-term potential and robustness. In addition, SMAF also exhibits competitive adaptability and convergence stability across the remaining test problems, further confirming its overall superiority in handling high-dimensional tri-objective optimization problems.

Fig. 12 illustrates the population distributions obtained by SMAF and the comparison algorithms on the three-objective test problems SMOP1, SMOP2, SMOP3, SMOP5, SMOP6, and SMOP7 with 500 decision variables. As can be seen, the solution sets generated by SMAF exhibit a more uniform distribution in the objective space and achieve better convergence towards the true Pareto front compared to the other algorithms.

In addition, we investigated the performance of SMAF and the compared algorithms on SMOP test functions with varying sparsity levels. Fig. 13 illustrates the IGD results obtained by SMAF and its competitors on eight SMOP problems, each with 500 decision variables and sparsity levels ranging from 0.1 to 0.9. As the sparsity increases, the performance of all eight algorithms generally declines. Except for the SMOP4 problem, SMAF still achieves better IGD performance than the other compared algorithms in most cases. This demonstrates its adaptability and robustness across varying sparsity structures.

4.4. Results on real-world applications

In this section, the proposed SMAF algorithm is compared with several peer algorithms on practical application problems. Table 4 specifically presents the HV performance of SMAF versus two representative real-coded algorithms, S-NSGA-II and S-ECSO, on the sparse signal reconstruction task. It can be observed that SMAF consistently achieves better HV values across all test instances, attaining the best average and the smallest standard deviation in each case. This suggests that SMAF tends to perform well in both solution quality and stability. On larger-scale problems such as SR2 and SR3, SMAF maintains its lead, with performance advantages increasing with problem dimensionality.

Given that S-NSGA-II and S-ECSO are originally designed for real-coded optimization, applying them directly to binary-coded problems is not considered appropriate. Moreover, different studies adopt varying strategies to adapt such algorithms to binary-coded problems, yet there is currently no widely accepted standard for these modifications. As a result, such adaptations may introduce additional sources of variation and uncertainty, making fair comparisons difficult. Therefore, these two algorithms are not included in the binary-coded problem experiments. Instead, algorithms specifically designed for or natively applicable to binary-coded problems are selected as baselines for comparison in those scenarios.

Table 5 presents the performance comparison of the SMAF algorithm against multiple competing algorithms in practical application scenarios. SMAF is evaluated on multiple test datasets, including sparse signal reconstruction (SR), feature selection (FS), critical node detection (CN), pattern mining problem (PM) and generally achieves competitive results relative to other established methods. For all twelve real-world test problems, SMAF achieves the best average HV values with generally smaller standard deviations, indicating superior and more stable performance. Compared to AGSEA, SMAF shows more pronounced improvements on high-dimensional problems (e.g., a 3.38 % gain on SR3), while achieving smaller yet consistent gains in near-optimal problems (e.g., FS3 and CN3). Compared with other algorithms, SMAF demonstrates competitive performance, with average HV values across the nine real-world test problems generally at least comparable to those of the other methods. From the problem classification perspective, SMAF's advantage becomes more prominent with increasing dimensionality (SR series dimensions from 1024 to 5120, with improvements from 1.56 % to 3.38 %). In medium-dimensional problems (FS and CN series) and near-optimal scenarios, SMAF shows relatively consistent performance across problems of all dimensions, indicating its reliability in a range of optimization contexts.

In the PM series of experiments, we evaluated the algorithms on sparse multi-objective optimization problems with varying decision variable dimensions. As the dimension increased from 500 to 2000, the performance of all algorithms varied. SMAF consistently achieved the best or near-best results across all three problems, demonstrating stable performance. SparseEA obtained results comparable to SMAF in all cases, indicating a certain level of competitiveness. AGSEA, MSKEA, and MOEAPSL performed slightly worse than SMAF. Overall, SMAF exhibits strong stability and adaptability in handling high-dimensional sparse multi-objective optimization problems.

Fig. 14 illustrates the population distributions of the median HV values obtained by SMAF and its competitors on SR1, CN1, and FS1. Specifically, SMAF improves the HV value by approximately 15.3 % over MSKEA on the SR1 problem and by about 0.45 % over AGSEA on the FS1 problem. Moreover, SMAF generally performs comparably to or slightly better than the competing algorithms on other test problems, with most improvements ranging from 0.1 % to 1.5 %, showing relatively consistent performance in convergence and diversity.

4.5. Comparison with other DRL-Based optimizers

To further validate the effectiveness of the proposed SMAF framework, it is compared with several representative deep reinforcement learning (DRL)-based multi-objective optimizers on a suite of sparse benchmark problems. The compared algorithms include DRLOS-EMCMO Ming et al. (2024) and MOEA/D-DQN Tian et al. (2022b), which represent distinct RL paradigms such as actor–critic and value-based frameworks.

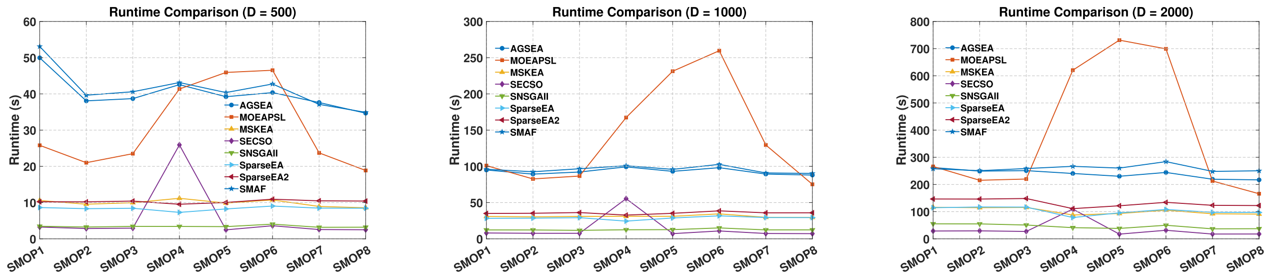

Fig. 15. Runtime comparison of different algorithms on various test problems. D = 500, 1000, 2000.

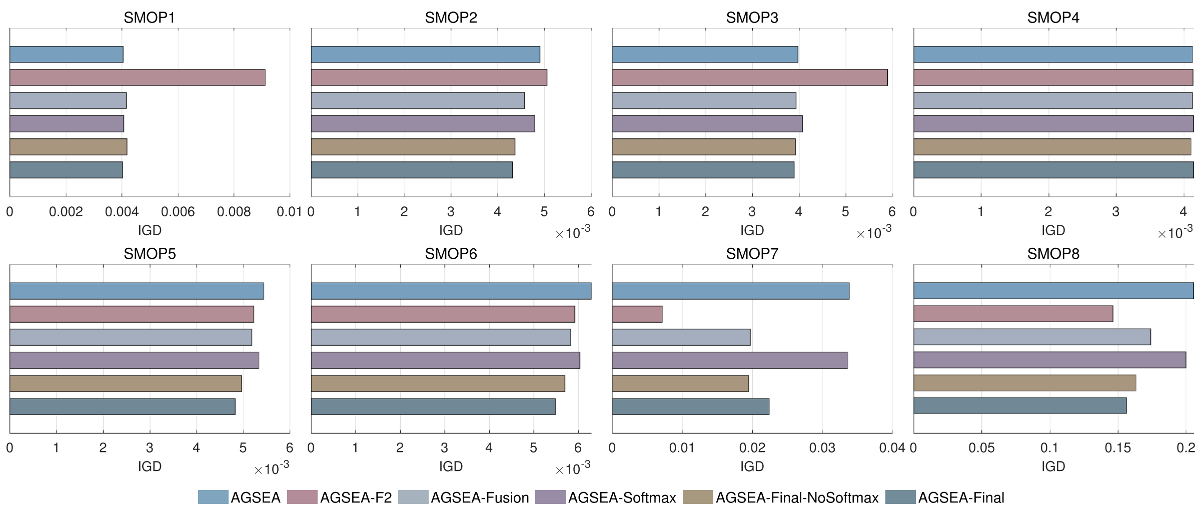

Fig. 16. IGD performance of AGSEA and its variants on bi-objective SMOP1-SMOP8 with 500 decision variables.

MOEA/D-DQN Tian et al. (2022b) and DRLOS-EMCMO Ming et al. (2024) are two representative DRL-based multi-objective evolutionary algorithms that integrate reinforcement learning into the optimization process. MOEA/D-DQN adopts a value-based Deep Q-Network (DQN) to adaptively select subproblems or variation operators within the MOEA/D framework, thereby improving the balance between exploration and exploitation. DRLOS-EMCMO, on the other hand, employs a DQN-driven operator selection strategy for constrained multi-objective optimization. It maintains an experience pool to store state–action–reward information, enabling adaptive learning of operator effectiveness during evolution.

To ensure a fair comparison under sparse conditions, the two algorithms are enhanced with the sparsity-handling mechanism of SparseEA. The resulting improved versions, DRLOS-EMCMO_Sparse and MOEA/D-DQN_Sparse, are designed to better handle large-scale sparse multi-objective optimization problems. All algorithms are tested under the same parameter settings and stopping criteria, and the performance is evaluated using the Inverted Generational Distance (IGD) metric. The statistical results on the bi-objective SMOP test problems are summarized in Table 7, while the results on the tri-objective SMOP test problems are presented in Table 8. These results collectively demonstrate the performance of the proposed algorithm across problems with different numbers of objectives.

Since DRLOS-EMCMO and MOEA/D-DQN are not originally designed for large-scale optimization, an additional $D = 512$ instance of the SR problem was tested to evaluate their performance, while the SMOP test problems are only considered at $D = 1000$.

Table 9 summarizes the performance of the algorithms on the SR problem in terms of the HV metric. It can be observed that SMAF consistently achieves the highest HV values across all dimensions ($D = 512, 1024, 2048$), with relatively small standard deviations, indicating stable performance. Specifically, for $D = 512$, SMAF improves the HV by approximately 9 % over DRLOS-EMCMO and around 7.6 % over MOEA/D-DQN. In comparison, DRLOS-EMCMO and its sparse variant exhibit lower HV values, which further decrease as the problem dimension increases, suggesting limited search capability in high-dimensional settings. The MOEA/D-DQN variants achieve intermediate HV values, showing that reinforcement learning guidance provides some improvement, but still falls short of SMAF. Overall, SMAF demonstrates a clear advantage in both effectiveness and robustness on the SR problem.

The experimental results indicate that on the SMOP1–SMOP8 bi-objective problems, SMAF achieves the lowest IGD values on most instances, except for SMOP4 and SMOP5, where MOEA/D-DQN_Sparse performs slightly better. On the three-objective problems, SMAF attains the best IGD on all instances. In contrast, the original MOEA/D-DQN and DRLOS-EMCMO algorithms exhibit consistently higher IGD values, and although the incorporation of sparsity strategies improves their performance, they still fail to surpass SMAF. As the number of objectives increases, the performance of the other algorithms deteriorates more noticeably, while SMAF maintains strong stability and robustness, demonstrating its clear advantage in large-scale sparse multi-objective optimization.

4.6. Computational complexity

The computational complexity of SMAF mainly depends on the problem dimension D, the population size N, and the number of generations T. In the initialization stage, the hybrid strategy (`FitnessCal` and `RealInit1`) requires $O(D^2 + ND)$ due to variable-wise fitness estimation and population construction. During each generation, the major computational costs arise from: (1) environmental selection $O(N^2)$, (2) evolutionary operators (including mask adjustment, crossover, and mutation) $O(ND)$, and (3) fitness evaluation of offspring $O(ND)$. Thus, the

Table 6
IGD values obtained by AGSEA, AGSEA-F2, AGSEA-Fusion, AGSEA-Softmax, AGSEA-Final-NoSoftmax, and AGSEA-Final on bi-objective SMOP1–SMOP8 with 500 decision variables. The best result in each row is highlighted.

Problem	D	AGSEA	AGSEA-F2	AGSEA-Fusion	AGSEA-Softmax	AGSEA-Final- NoSoftmax	AGSEA-Final
SMOP1	500	4.0410e-3 (7.05e-5) =	9.1104e-3 (5.57e-3) -	4.1551e-3 (2.46e-4) -	4.0717e-3 (1.58e-4) =	4.1830e-3 (3.82e-4) -	**4.0140e-3 (7.22e-5)**
SMOP2	500	4.9007e-3 (1.34e-3) -	5.0518e-3 (1.40e-3) -	4.5711e-3 (4.94e-4) -	4.7904e-3 (8.58e-4) -	4.3654e-3 (3.74e-4) =	**4.3092e-3 (3.81e-4)**
SMOP3	500	3.9762e-3 (2.46e-4) -	5.8974e-3 (7.82e-3) -	3.9315e-3 (5.23e-5) -	4.0724e-3 (4.19e-4) -	3.9230e-3 (1.46e-4) =	**3.8958e-3 (9.35e-5)**
SMOP4	500	4.1312e-3 (6.86e-5) =	4.1347e-3 (6.23e-5) =	4.1294e-3 (7.84e-5) =	4.1462e-3 (8.78e-5) =	**4.1033e-3 (7.46e-5)** +	4.1302e-3 (6.93e-5)
SMOP5	500	5.4374e-3 (3.02e-4) -	5.2276e-3 (2.78e-4) -	5.1850e-3 (3.51e-4) -	5.3289e-3 (2.67e-4) -	4.9624e-3 (2.14e-4) -	**4.8317e-3 (2.22e-4)**
SMOP6	500	6.2892e-3 (4.55e-4) -	5.9204e-3 (2.87e-4) -	5.8315e-3 (3.63e-4) -	6.0344e-3 (4.75e-4) -	5.6992e-3 (2.83e-4) -	**5.4836e-3 (2.26e-4)**
SMOP7	500	3.3834e-2 (8.22e-3) -	**7.0904e-3 (5.50e-3)** =	1.9696e-2 (1.89e-2) =	3.3588e-2 (9.53e-3) -	1.9437e-2 (2.65e-2) =	2.2368e-2 (2.76e-2)
SMOP8	500	2.0552e-1 (1.45e-2) -	**1.4642e-1 (1.18e-2)** =	1.7398e-1 (3.96e-2) -	1.9993e-1 (1.50e-2) -	1.6300e-1 (3.82e-2) =	1.5613e-1 (3.22e-2)
+/-/=		0/6/2	0/5/3	0/6/2	0/6/2	1/3/4	

Table 7
The mean and standard deviation of IGD results obtained by SMAF and its comparison algorithms on the SMOP1-SMOP8 bi-objective benchmark problems, where the best result in each row is highlighted.

Problem	D	MOEA/D-DQN	MOEA/D-DQN_Sparse	DRLOS-EMCMO	DRLOS-EMCMO_Sparse	SMAF
SMOP1	1000	6.1883e-1 (4.83e-2) -	7.3862e-2 (7.56e-4) -	4.9831e-1 (2.41e-2) -	6.9838e-2 (4.47e-3) -	**4.0478e-3 (6.56e-5)**
SMOP2	1000	1.5444e+0 (1.64e-1) -	1.5683e-1 (1.12e-3) -	1.4794e+0 (4.86e-2) -	1.5273e-1 (7.69e-3) -	**4.4260e-3 (1.29e-4)**
SMOP3	1000	1.4290e+0 (1.40e-1) -	7.6321e-2 (8.35e-4) -	1.2819e+0 (1.20e-1) -	7.0081e-1 (4.48e-1) -	**3.9679e-3 (6.99e-5)**
SMOP4	1000	8.5606e-1 (8.21e-2) -	**3.9472e-3 (9.54e-7)** +	7.8554e-1 (1.44e-2) -	4.3849e-3 (2.12e-4) -	4.1348e-3 (7.80e-5)
SMOP5	1000	4.2849e-1 (1.03e-2) -	**4.3696e-3 (1.37e-4)** +	4.2782e-1 (3.76e-3) -	3.0317e-2 (8.38e-3) -	4.6126e-3 (1.32e-4)
SMOP6	1000	2.0522e-1 (1.77e-2) -	3.9742e-2 (5.53e-4) -	1.6821e-1 (4.95e-3) -	5.1000e-2 (4.53e-3) -	**4.6740e-3 (1.79e-4)**
SMOP7	1000	6.3636e-1 (8.29e-2) -	2.2425e-1 (1.27e-3) -	5.6495e-1 (4.84e-2) -	1.9515e-1 (7.89e-3) -	**4.6113e-3 (3.80e-4)**
SMOP8	1000	2.5260e+0 (3.87e-1) -	5.0612e-1 (4.92e-3) -	2.4972e+0 (6.64e-2) -	4.7219e-1 (2.85e-2) -	**1.4521e-1 (8.28e-3)**
+/-/=		0/8/0	2/6/0	0/8/0	0/8/0	

overall time complexity can be summarized as (see Eq. (15)):

$$O(D^2 + T(N^2 + ND)) \tag{15}$$

When $D \gg N$, the complexity is dominated by ND, while for large population sizes, the N^2 term becomes the main bottleneck. Compared with AGSEA, SMAF introduces additional initialization cost from `RealInit1` but does not change the asymptotic complexity.

In terms of actual running time, SMAF exhibits moderate computational cost among the seven compared algorithms. It is often slower than AGSEA, but the difference remains within an acceptable range. MOEA/PSL demonstrates higher runtime overheads in many cases, although it may outperform SMAF in some scenarios. Runtime measurements across problems with 500, 1000, and 2000 decision variables (see Fig. 15) indicate that SMAF scales consistently with theoretical complexity, increasing steadily with problem dimension while remaining comparable to AGSEA. This suggests that the reinforcement learning module introduces additional overhead without fundamentally compromising scalability.

Despite these encouraging results, a notable limitation of SMAF is the extra computational cost associated with the reinforcement learning module. These findings highlight a trade-off between computational efficiency and solution quality: the reinforcement learning module increases runtime but contributes to improved optimization performance. Future work may focus on lightweight learning strategies or parallel implementations to further reduce this overhead.

5. Ablation experiment

5.1. Ablation experiment design and analysis on fitness strategies and action space

To evaluate the impact of different fitness evaluation strategies and action space designs on the performance of the AGSEA algorithm, the following variants were developed for ablation experiments:

- **AGSEA**: The original algorithm, where the action space consists of three actions. Among them, Action 1 selects decision variables based on the traditional prior information *F1*, while Actions 2 and 3 remain unchanged. A greedy strategy is adopted for action selection.
- **AGSEA-F2**: Based on AGSEA, this version replaces the original prior information *F1* with a prior information *F2*, constructed through variable interval partitioning and a fixed-value evaluation scheme. The action space remains unchanged.
- **AGSEA-Fusion**: Based on AGSEA, this version introduces *F2* and replaces the original random-based feedback fitness action with a weighted combination of *F1* and *F2*. The action space remains unchanged, and other actions are unchanged.
- **AGSEA-Softmax**: Builds on the original AGSEA to verify the effect of replacing the greedy strategy with a Softmax-based probabilistic action selection.
- **AGSEA-Final-NoSoftmax**: Based on AGSEA-Fusion, this version adds a new action derived from *F2*, thus expanding the action space to support multiple fitness evaluation strategies. However, the action selection still follows the greedy strategy, without employing the Softmax mechanism.
- **AGSEA-Final**: This final version builds upon AGSEA-Final-NoSoftmax by introducing the Softmax strategy in place of the greedy mechanism, allowing for probabilistic action selection based on action scores. It integrates improvements in new prior information, action space expansion, and selection strategies, and represents the complete integration of the proposed improvements.

AGSEA-F2 assesses the impact of the new prior information (F2) on variable selection performance. AGSEA-Fusion incorporates a weighted combination of F1 and F2 to evaluate the enhancement of feedback signal guidance. AGSEA-Softmax introduces the Softmax strategy to investigate its effect on increasing diversity and exploration during action selection. AGSEA-Final-No-Softmax expands the action space while retaining the greedy selection strategy to examine the effectiveness of multiple fitness evaluation strategies. Finally, AGSEA-Final combines fitness function optimization, action space expansion, and probabilistic action selection to comprehensively evaluate the overall performance of the proposed method.

The experimental results, as presented in Table 6 and 10, cover two mainstream performance indicators: IGD and HV. In terms of fitness

Table 8
The mean and standard deviation of IGD results obtained by SMAF and its comparison algorithms on the SMOP1-SMOP8 three-objective benchmark problems, where the best result in each row is highlighted.

Problem	D	MOEA/D-DQN	MOEA/D-DQN_Sparse	DRLOS-EMCMO	DRLOS-EMCMO_Sparse	SMAF
SMOP1	1000	5.8988e-1 (4.77e-2) -	8.0482e-2 (1.38e-3) -	6.0925e-1 (1.67e-2) -	1.3087e-1 (7.94e-3) -	**4.2502e-2 (4.26e-4)**
SMOP2	1000	1.5865e+0 (1.04e-1) -	1.4425e-1 (3.21e-3) -	1.7552e+0 (2.29e-2) -	1.9241e-1 (1.23e-2) -	**5.3650e-2 (1.81e-3)**
SMOP3	1000	1.3109e+0 (9.91e-2) -	2.4027e-1 (1.72e-1) -	1.5064e+0 (4.40e-2) -	1.0451e+0 (1.26e-1) -	**4.3775e-2 (7.98e-4)**
SMOP4	1000	6.0923e-1 (1.41e-2) -	3.6417e-2 (4.19e-2) -	6.4806e-1 (8.99e-3) -	4.5699e-2 (4.35e-3) -	**3.0576e-2 (6.13e-4)**
SMOP5	1000	2.7878e-1 (7.20e-3) -	6.8461e-2 (8.70e-2) -	3.1904e-1 (1.01e-2) -	1.0090e-1 (1.43e-2) -	**3.1921e-2 (8.41e-4)**
SMOP6	1000	1.5067e-1 (7.10e-3) -	5.1281e-2 (4.61e-2) -	1.8812e-1 (8.05e-3) -	1.1176e-1 (1.51e-2) -	**3.1497e-2 (5.89e-4)**
SMOP7	1000	8.1833e-1 (1.01e-1) -	2.3688e-1 (1.33e-3) -	7.8725e-1 (4.75e-2) -	2.7861e-1 (9.86e-3) -	**6.2912e-2 (9.03e-4)**
SMOP8	1000	2.8977e+0 (3.21e-1) -	5.1727e-1 (1.83e-3) -	3.7018e+0 (1.75e-2) -	5.7296e-1 (2.58e-2) -	**2.1573e-1 (1.14e-2)**
+/-/=		0/8/0	0/8/0	0/8/0	0/8/0	

Table 9
The mean and standard deviation of HV results obtained by SMAF and its comparison algorithms on the SMOP1-SMOP8 three-objective benchmark problems, where the best result in each row is highlighted.

Problem	D	DRLOS-EMCMO	DRLOS-EMCMO_Sparse	MOEA/D-DQN	MOEA/D-DQN_Sparse	SMAF
	512	3.4218e-1 (1.36e-2) -	2.4454e-1 (2.25e-2) -	3.4806e-1 (1.35e-2) -	3.4973e-1 (1.08e-2) -	**3.7681e-1 (2.66e-3)**
SR	1024	3.3651e-1 (1.59e-2) -	2.8454e-1 (1.33e-2) -	3.4087e-1 (1.43e-2) -	3.3867e-1 (7.99e-3) -	**3.9387e-1 (1.13e-3)**
	2048	3.0020e-1 (2.04e-2) -	2.4473e-1 (1.69e-2) -	3.0678e-1 (1.07e-2) -	3.0950e-1 (8.62e-3) -	**3.7910e-1 (9.27e-4)**
+/-/=		0/3/0	0/3/0	0/3/0	0/3/0	

Table 10
HV values obtained by AGSEA, AGSEA-F2, AGSEA-Fusion, AGSEA-Softmax, AGSEA-Final-NoSoftmax, and AGSEA-Final on bi-objective SMOP1–SMOP8 with 500 decision variables. The best result in each row is highlighted.

Problem	D	AGSEA	AGSEA-F2	AGSEA-Fusion	AGSEA-Softmax	AGSEA-Final- NoSoftmax	AGSEA-Final
SMOP1	500	5.8128e-1 (1.55e-4) =	5.7348e-1 (7.73e-3) -	5.8105e-1 (5.45e-4) -	5.8122e-1 (3.76e-4) =	5.8100e-1 (8.05e-4) -	**5.8134e-1 (1.60e-4)**
SMOP2	500	5.7960e-1 (2.16e-3) -	5.7935e-1 (2.30e-3) -	5.8009e-1 (9.10e-4) -	5.7976e-1 (1.51e-3) -	5.8055e-1 (6.93e-4) =	**5.8066e-1 (7.48e-4)**
SMOP3	500	5.8137e-1 (5.66e-4) -	5.7873e-1 (1.04e-2) =	5.8149e-1 (1.17e-4) -	5.8117e-1 (9.41e-4) =	5.8150e-1 (3.68e-4) =	**5.8158e-1 (2.36e-4)**
SMOP4	500	8.1868e-1 (4.45e-5) =	8.1868e-1 (4.88e-5) =	8.1869e-1 (5.88e-5) =	8.1867e-1 (6.10e-5) =	**8.1870e-1 (5.14e-5)** +	8.1868e-1 (5.02e-5)
SMOP5	500	8.1557e-1 (4.84e-4) -	8.1595e-1 (4.66e-4) -	8.1596e-1 (5.74e-4) -	8.1575e-1 (4.69e-4) -	8.1643e-1 (3.75e-4) -	**8.1668e-1 (4.34e-4)**
SMOP6	500	8.1432e-1 (6.26e-4) -	8.1482e-1 (4.09e-4) -	8.1497e-1 (5.10e-4) -	8.1467e-1 (6.71e-4) -	8.1518e-1 (4.23e-4) -	**8.1546e-1 (3.37e-4)**
SMOP7	500	3.0497e-1 (1.16e-2) -	**3.4228e-1 (8.22e-3)** +	3.2409e-1 (2.66e-2) =	3.0475e-1 (1.38e-2) -	3.2466e-1 (3.80e-2) =	3.2047e-1 (3.94e-2)
SMOP8	500	1.0945e-1 (1.07e-2) -	**1.5832e-1 (1.11e-2)** =	1.3658e-1 (3.05e-2) -	1.1341e-1 (1.11e-2) -	1.4613e-1 (2.95e-2) =	1.5170e-1 (2.56e-2)
+/-/=		0/6/2	1/4/3	0/6/2	0/5/3	1/3/4	

Table 11
IGD values obtained by AGSEA and AGSEA-SG on bi-objective SMOP1–SMOP8 with 500 decision variables. The best result in each row is highlighted. '+', '-', and '≈' indicate that the result by another MOEA is better than, worse than, or comparable to that obtained by AGSEA-SG.

Problem	D	AGSEA	AGSEA-SG
SMOP1	500	4.0410e-3 (7.05e-5) -	**3.9963e-3 (8.72e-5)**
SMOP2	500	4.9007e-3 (1.34e-3) -	**4.3602e-3 (2.52e-4)**
SMOP3	500	**3.9762e-3** (2.46e-4) =	4.0895e-3 (3.29e-4)
SMOP4	500	4.1312e-3 (6.86e-5) =	**4.1169e-3** (7.88e-5)
SMOP5	500	5.4374e-3 (3.02e-4) -	**5.2840e-3** (3.93e-4)
SMOP6	500	6.2892e-3 (4.55e-4) -	**5.2198e-3** (2.20e-4)
SMOP7	500	3.3834e-2 (8.22e-3) -	**5.2131e-3** (8.29e-4)
SMOP8	500	2.0552e-1 (1.45e-2) -	**1.4146e-1** (1.02e-2)
+/-/=		0/6/2	

Table 12
HV values obtained by AGSEA and AGSEA-SG on bi-objective SMOP1–SMOP8 with 500 decision variables. The best result in each row is highlighted.

Problem	D	AGSEA	AGSEA-SG
SMOP1	500	5.8128e-1 (1.55e-4) -	**5.8137e-1 (1.73e-4)**
SMOP2	500	5.7960e-1 (2.16e-3) -	**5.8051e-1 (4.82e-4)**
SMOP3	500	**5.8137e-1** (5.66e-4) =	5.8108e-1 (8.06e-4)
SMOP4	500	8.1868e-1 (4.45e-5) =	**8.1870e-1** (5.19e-5)
SMOP5	500	8.1557e-1 (4.84e-4) =	**8.1579e-1** (6.40e-4)
SMOP6	500	8.1432e-1 (6.26e-4) -	**8.1595e-1** (3.72e-4)
SMOP7	500	3.0497e-1 (1.16e-2) -	**3.4578e-1** (1.49e-3)
SMOP8	500	1.0945e-1 (1.07e-2) -	**1.6406e-1** (9.58e-3)
+/-/=		0/5/3	

function optimization, AGSEA-F2 employs a newly constructed prior information (F2), based on variable partitioning and correlation analysis, to replace the original Fitness1. Although this method achieves comparable performance to the final version on three test problems, it performs worse than the baseline AGSEA on SMOP1 and SMOP2, indicating that a single evaluation mechanism struggles to balance global exploration and local exploitation. Regarding HV, AGSEA-F2 falls short of the fi-

Table 13
IGD performance comparison of different AGSEA variants on benchmark sparse multi-objective problems. The best result in each row is highlighted.

Problem	D	AGSEA	AGSEA-MA	AGSEA-SG	AGSEA-HF
SMOP1	500	4.0410e-3 (7.05e-5) -	4.0140e-3 (7.22e-5) -	3.9963e-3 (8.72e-5) =	**3.9619e-3 (8.36e-5)**
SMOP2	500	4.9007e-3 (1.34e-3) -	4.3092e-3 (3.81e-4) =	4.3602e-3 (2.52e-4) -	**4.1920e-3 (1.28e-4)**
SMOP3	500	3.9762e-3 (2.46e-4) =	**3.8958e-3 (9.35e-5)** +	4.0895e-3 (3.29e-4) =	3.9621e-3 (1.38e-4)
SMOP4	500	4.1312e-3 (6.86e-5) =	4.1465e-3 (6.91e-5) =	**4.1169e-3 (7.88e-5)** =	4.1370e-3 (4.54e-5)
SMOP5	500	5.4374e-3 (3.02e-4) -	4.8317e-3 (2.22e-4) =	5.2840e-3 (3.93e-4) -	**4.7501e-3 (2.05e-4)**
SMOP6	500	6.2892e-3 (4.55e-4) -	5.4836e-3 (2.26e-4) -	5.2198e-3 (2.20e-4) -	**4.7966e-3 (1.99e-4)**
SMOP7	500	3.3834e-2 (8.22e-3) -	2.2368e-2 (2.76e-2) -	5.2131e-3 (8.29e-4) -	**4.4017e-3 (9.68e-5)**
SMOP8	500	2.0552e-1 (1.45e-2) -	1.5613e-1 (3.22e-2) =	**1.4146e-1 (1.02e-2)** =	1.4214e-1 (9.32e-3)
+/-/=		0/6/2	1/3/4	0/4/4	

Table 14
HV performance comparison of different AGSEA variants on benchmark sparse multi-objective problems. The best result in each row is highlighted.

Problem	D	AGSEA	AGSEA-MA	AGSEA-SG	AGSEA-HF
SMOP1	500	5.8128e-1 (1.55e-4) -	5.8134e-1 (1.60e-4) -	5.8137e-1 (1.73e-4) =	**5.8145e-1 (1.71e-4)**
SMOP2	500	5.7960e-1 (2.16e-3) -	5.8066e-1 (7.48e-4) =	5.8051e-1 (4.82e-4) -	**5.8090e-1 (2.70e-4)**
SMOP3	500	5.8137e-1 (5.66e-4) =	**5.8158e-1 (2.36e-4)** +	5.8108e-1 (8.06e-4) =	5.8141e-1 (3.39e-4)
SMOP4	500	8.1868e-1 (4.45e-5) =	8.1867e-1 (4.88e-5) =	**8.1870e-1 (5.19e-5)** =	8.1868e-1 (3.50e-5)
SMOP5	500	8.1557e-1 (4.84e-4) -	8.1668e-1 (4.34e-4) =	8.1579e-1 (6.40e-4) -	**8.1680e-1 (3.56e-4)**
SMOP6	500	8.1432e-1 (6.26e-4) -	8.1546e-1 (3.37e-4) -	8.1595e-1 (3.72e-4) -	**8.1668e-1 (3.33e-4)**
SMOP7	500	3.0497e-1 (1.16e-2) -	3.2047e-1 (3.94e-2) -	3.4578e-1 (1.49e-3) -	**3.4656e-1 (2.11e-4)**
SMOP8	500	1.0945e-1 (1.07e-2) -	1.5170e-1 (2.56e-2) =	**1.6406e-1 (9.58e-3)** =	1.6351e-1 (8.91e-3)
+/-/=		0/6/2	1/3/4	0/4/4	

nal version on more than four problems, further confirming its limited capability in delivering expressive feedback signals.

AGSEA-Fusion enhances feedback signal expressiveness by weighting and combining Fitness1 and Fitness2. It achieves better IGD results than the original AGSEA across most problems from SMOP2 to SMOP8, with particularly notable improvements on SMOP7 and SMOP8. This demonstrates that the fusion strategy contributes positively to both diversity preservation and convergence. However, in terms of HV, AGSEA-Fusion still underperforms the final version in most cases, suggesting room for improvement in approximating the Pareto front boundaries.

For action selection, AGSEA-Softmax introduces a probabilistic Softmax mechanism to replace greedy selection, performing exponential mapping and normalization of action scores to enable probability-based sampling. Although this theoretically promotes policy diversity, the results across both IGD and HV indicate that it fails to achieve improvements on six of the test problems, suggesting that the Softmax strategy, when used in isolation, is insufficient and must be integrated with other enhancements to realize its full potential.

AGSEA-Final-No-Softmax expands the action space by incorporating multiple fitness-based action feedback schemes while retaining the original greedy mechanism. This variant outperforms AGSEA in both IGD and HV metrics on SMOP2, SMOP5, and SMOP6, verifying the positive effect of action space expansion. However, its performance is still inferior to the final version, especially on SMOP1 and SMOP2, indicating that the Softmax mechanism plays a critical compensatory role in guiding action sampling under complex scenarios. These findings support the theoretical and empirical necessity of combining Softmax with other strategies in the final design.

The final version, AGSEA-Final, integrates new prior information, feedback fusion, action space expansion, and Softmax-based sampling. It achieves the best or statistically equivalent performance across all eight test problems. In terms of IGD, AGSEA-Final outperforms the original AGSEA on six problems (e.g., achieving a 33.9 % improvement on SMOP7 and 24 % on SMOP8), while standard deviations decrease by 30 % to 50 % across most cases. Regarding HV, AGSEA-Final also demonstrates superiority on multiple problems, notably achieving stable improvements over AGSEA on SMOP5, SMOP6, and SMOP8. These results collectively highlight the dual advantages of improved diversity and convergence quality, validating the effectiveness and general applicability of the proposed multi-strategy collaborative mechanism.

In addition, Fig. 16 further visualizes the mean IGD performance of each algorithm on the benchmark problems. The results clearly show that AGSEA-Final outperforms all other variants on several problems, with particularly notable improvements on SMOP7 and SMOP8. This demonstrates the effectiveness of the proposed multi-strategy collaborative mechanism in enhancing both convergence and robustness. And Fig. 17 provides a heatmap of IGD-based rankings for the six algorithms across the eight SMOP problems. Each cell displays the ranking position, with darker colors representing better performance. AGSEA-Final achieves the best performance on 5 out of the 8 test problems. Although its performance on the SMOP4 problem is relatively weaker, the overall IGD values across all algorithms on this problem are very close, with negligible differences. Therefore, the performance on SMOP4 does not substantially undermine the overall competitiveness of AGSEA-Final.

Further analysis reveals that AGSEA-Final not only achieves superior mean IGD values on most problems but also demonstrates smaller standard deviations, indicating stronger stability and robustness. Specifically, it achieves the best results on SMOP1, SMOP3, SMOP5, SMOP6, and SMOP8, which cover diverse distribution characteristics and varying problem complexities. This highlights the good generalization ability of the proposed algorithm.

5.2. Ablation study on sparsity-guided action space design

To evaluate the performance improvement contributed by the newly introduced sparse guidance adjustment module, we conducted ablation experiments. In these experiments, we compared the algorithm incorporating the sparse guidance mechanism with a baseline version that excludes this module.

Specifically, the sparse guidance module dynamically adjusts the activation ratio of variables in the mask to maintain the sparsity level of individuals within a reasonable range observed in non-dominated solutions. It also integrates fitness information to guide variable selection, thereby enhancing search efficiency and solution quality. The baseline version is the original algorithm without the sparse guidance adjustment module, while all other operators and parameters remain unchanged.

Table 15
IGD performance comparison of SMAF strategies on SMOP problems (D = 500). Bold values indicate the best results.

Problem	D	SMAF_Early_Fast	SMAF_Late_Fast	SMAF_Mid_Fast	SMAF_Mid_Slow	SMAF
SMOP1	500	3.9740e-3 (7.47e-5) =	**3.9619e-3 (7.32e-5)** =	3.9848e-3 (8.08e-5) =	3.9819e-3 (7.19e-5) =	3.9875e-3 (1.22e-4)
SMOP2	500	4.2316e-3 (2.06e-4) =	**4.1576e-3 (1.31e-4)** =	4.1849e-3 (1.54e-4) =	4.2046e-3 (1.42e-4) =	4.1842e-3 (1.55e-4)
SMOP3	500	3.9563e-3 (1.81e-4) =	4.0318e-3 (3.47e-4) =	**3.9041e-3 (5.41e-5)** =	3.9307e-3 (1.51e-4) =	3.9079e-3 (4.18e-5)
SMOP4	500	4.1422e-3 (5.93e-5) =	4.1401e-3 (7.44e-5) =	**4.1187e-3 (6.52e-5)** =	4.1228e-3 (7.12e-5) =	4.1345e-3 (6.45e-5)
SMOP5	500	4.9046e-3 (2.09e-4) =	**4.7670e-3 (1.93e-4)** =	4.7878e-3 (2.18e-4) =	4.8013e-3 (1.89e-4) =	4.8056e-3 (2.50e-4)
SMOP6	500	4.7918e-3 (2.17e-4) +	4.8496e-3 (2.09e-4) =	4.8415e-3 (2.45e-4) =	**4.7772e-3 (1.42e-4)** +	4.9256e-3 (2.40e-4)
SMOP7	500	4.8339e-3 (1.26e-3) =	**4.4249e-3 (1.29e-4)** =	4.5982e-3 (4.96e-4) =	4.5719e-3 (5.14e-4) =	4.6119e-3 (8.66e-4)
SMOP8	500	1.3998e-1 (9.98e-3) =	1.4197e-1 (9.32e-3) =	1.3932e-1 (7.68e-3) =	1.4193e-1 (8.54e-3) =	**1.3861e-1 (7.76e-3)**
+/-/=		1/0/7	0/0/8	1/0/7	0/0/8	

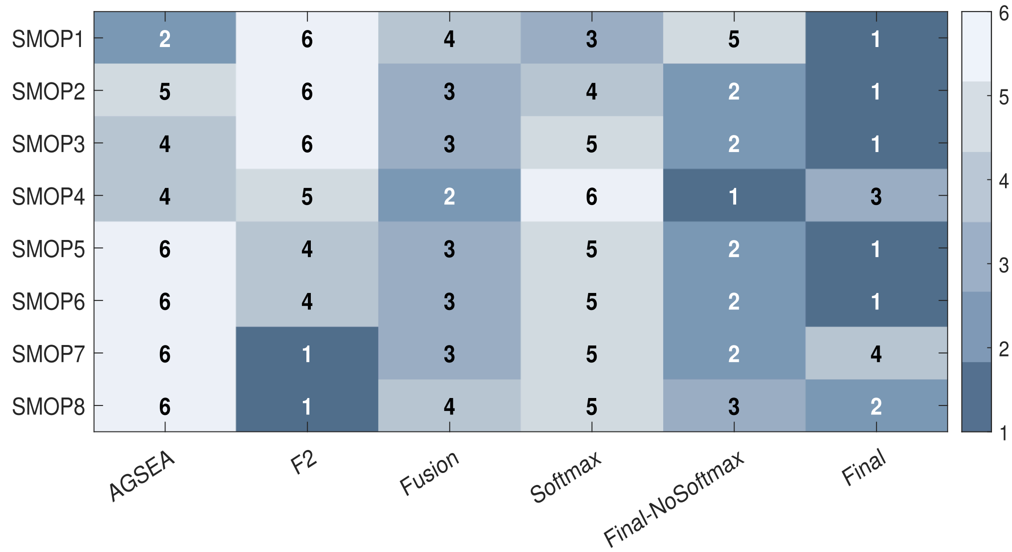

Fig. 17. Heatmap of IGD-based rankings for AGSEA and its variants on bi-objective SMOP1–SMOP8 problems (500 decision variables). For each problem (row), a lower rank indicates better performance, with darker colors representing better rankings.

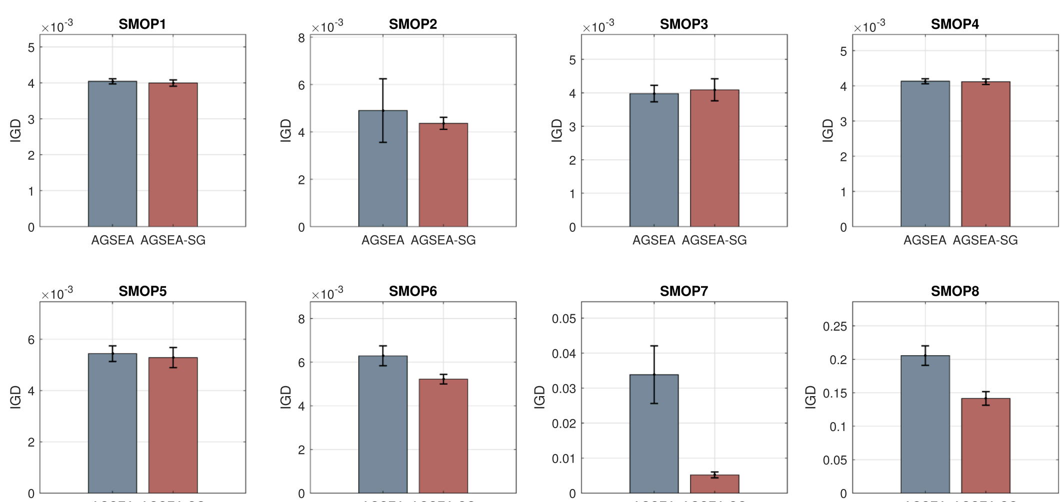

Fig. 18. IGD performance of AGSEA and AGSEA-SG on bi-objective SMOP1-SMOP8 with 500 decision variables.

By comparing these two versions, we assess the actual contribution of the sparse guidance module in controlling mask sparsity and improving evolutionary search performance.

Table 11 presents the IGD performance of AGSEA and AGSEA-SG on the SMOP benchmark problems. Fig. 18 further visualizes the average IGD performance of each algorithm on the benchmark problems. According to the analysis in Table 11, AGSEA-SG outperforms or matches AGSEA on most of the test problems. Specifically, among the eight test problems, AGSEA-SG achieves better IGD values on seven problems, with six of them performs better than AGSEA. Only on SMOP3 does AGSEA slightly outperform AGSEA-SG. In terms of statistical results, AGSEA-SG demonstrates six advantages, zero disadvantages, and two

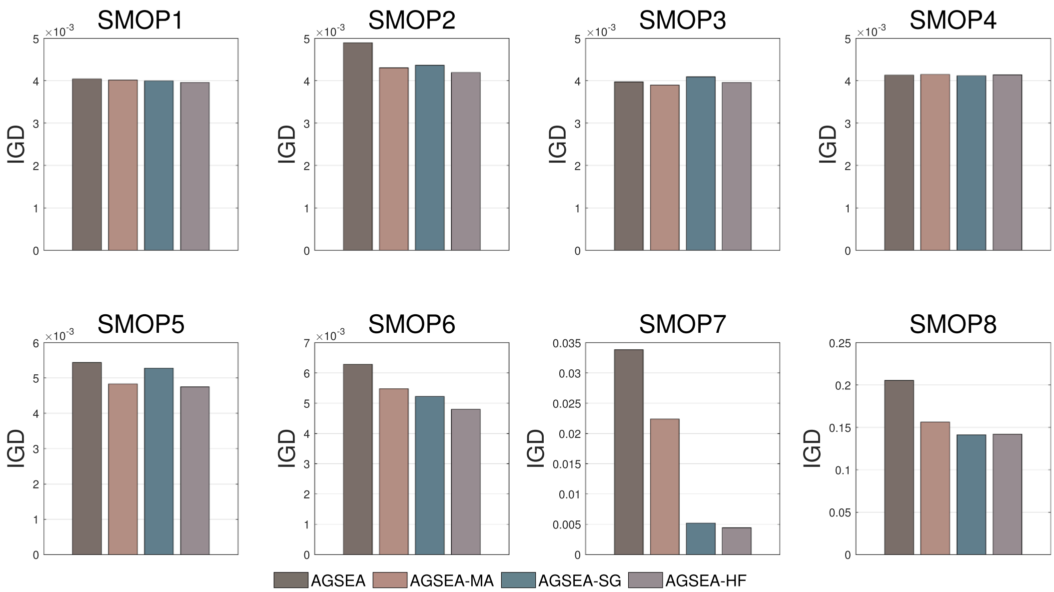

Fig. 19. IGD performance of AGSEA and its variants on bi-objective SMOP1–SMOP8 with 500 decision variables.

ties compared to AGSEA. These results indicate that introducing the sparse-guided mechanism significantly enhances the algorithm's ability to maintain convergence and diversity of the solution set.

In addition, Table 12 reports the performance comparison in terms of the HV metric. The results show that AGSEA-SG also exhibits superior HV values. Among the eight problems, AGSEA-SG outperforms AGSEA on five problems and ties on the remaining three, with no inferior performance observed. These findings further validate the effectiveness of AGSEA-SG in improving the overall solution quality. Notably, on SMOP7 and SMOP8, AGSEA-SG improves HV values by approximately 13.4 % and 49.9 %, respectively, indicating its enhanced ability to maintain convergence while improving population coverage in the objective space.

In summary, AGSEA-SG, obtained by integrating a sparse-guided adjustment module into AGSEA, demonstrates improved performance on both IGD and HV indicators. This enhancement contributes to the convergence speed and distribution quality of the solution set in high-dimensional multi-objective optimization problems, and may also improve the algorithm's adaptability to complex Pareto front structures.

5.3. Ablation experiment and synergistic effect analysis of fusion mechanism

To thoroughly investigate the individual contributions of the proposed modules and their combined effects, a series of ablation experiments are conducted on representative benchmark sparse multi-objective optimization problems. Specifically, we focus on evaluating the optimization capability of the sparsity-guided mechanism and the multi-action selection with fitness-driven adjustment, as well as whether their integration yields any synergistic benefits. These experiments aim to validate the effectiveness of each component in terms of convergence performance and solution diversity, and to further explore the potential complementarity between the modules.

The involved algorithm variants are defined as follows:

- **AGSEA-MA:** Also referred to as AGSEA-Final in Section 5.1, this version incorporates a multi-action selection strategy and fitness-driven adjustment mechanism, enhancing the adaptability of action execution.
- **AGSEA-SG:** This variant integrates the sparsity-guided mutation mechanism, which adaptively adjusts the variable selection strategy based on sparsity feedback (see Section 5.2).
- **AGSEA-HF:** This is the hybrid fusion version that combines both the sparsity-guided mechanism and the multi-action fitness-enhanced strategy. The goal is to exploit their complementary strengths and evaluate their synergistic effect.

The results presented in Tables 13 and 14 indicate that the final integrated version, which simultaneously incorporates sparsity-guided and multi-action selection mechanisms, achieves the best or nearly best IGD performance in 5 out of the 8 test problems. This demonstrates a strong complementarity and synergistic enhancement between the two modules in the search strategy. In addition, Fig. 19 further visualizes the average IGD performance of each algorithm on the benchmark problems. Meanwhile, in terms of the HV metric, the integrated version also performs well, maintaining solution set diversity and coverage across most problems. Although the integrated strategy shows slightly inferior performance to some individual modules on certain metrics for a few problems, overall, the integration improves the algorithm's adaptability and robustness, achieving a favorable balance between convergence and diversity, thereby exhibiting more stable and superior optimization performance.

5.4. Parameter sensitivity analysis

To systematically investigate the effect of training start time and network update frequency on SMAF performance, we define two key parameters: the training start ratio (TSR) and the update interval (UI). Four variants were designed based on different combinations of these parameters:

- **SMAF_Early_Fast** (TSR = 0.1, UI = 10): starts training early with frequent network updates, achieving fast convergence but potentially less stable.
- **SMAF_Late_Fast** (TSR = 0.5, UI = 10): starts training later with frequent updates, resulting in slower convergence and higher computational cost.
- **SMAF_Mid_Fast** (TSR = 0.25, UI = 5): mid-stage start with frequent updates, providing moderate convergence.
- **SMAF_Mid_Slow** (TSR = 0.25, UI = 20): mid-stage start with sparse updates, leading to slower yet more stable convergence.

From Table 15, most SMAF variants achieve IGD values comparable to the original SMAF across the majority of test instances. Specifically, for SMOP1–SMOP5 and SMOP7–SMOP8, the variants' IGD values are similar to the baseline, and the standard deviations indicate overlapping performance distributions, suggesting no significant differences among the strategies. The only notable exception is SMOP6, where both SMAF_Mid_Slow and SMAF_Early_Fast attain lower IGD values with relatively small standard deviations, indicating noticeable improvements. Overall, although some variants occasionally show slight advantages on specific instances, the overall performance differences are limited, and most variants perform similarly to the original SMAF.

These findings suggest that, under the tested settings, the SMAF algorithm exhibits low sensitivity to the variations of training start ratio (TSR) and network update interval (UI). Although lthough early or late training initiation and frequent or sparse network updates may affect convergence speed on individual problems, their influence on the final IGD values is limited. This indicates that the algorithm's overall convergence behavior and solution quality are robust with respect to these two parameters, and fine-tuning TSR and UI is unlikely to produce substantial performance gains for most SMOP instances.

6. Conclusions

This paper addresses the limitations of the AGSEA algorithm in generalization ability and sparsity control for large-scale sparse multi-objective optimization problems. We propose SMAF, a novel evolutionary optimization framework integrating deep reinforcement learning with structure-guided mechanisms. SMAF employs structure-aware sampling to capture dependencies among decision variables, a Softmax-based action selection strategy to enhance search diversity and stability, and an adaptive feedback mechanism to regulate sparsity. By dynamically selecting representative guiding vectors, it efficiently generates sparse and well-distributed solution sets.

Extensive experiments on benchmark problems and real-world applications–including sparse signal reconstruction, critical node detection, and high-dimensional feature selection–demonstrate that SMAF achieves competitive or superior performance while providing practical benefits. In sparse signal reconstruction, it balances sparsity and reconstruction accuracy, reducing measurement costs. For critical node detection, it identifies key nodes with minimal computational burden, supporting network design and robustness analysis. In feature selection, SMAF efficiently selects informative features to improve predictive performance and avoid overfitting. These results highlight SMAF's applicability to real-world sparse optimization tasks.

Overall, SMAF provides both an effective algorithmic solution and a broadly applicable framework bridging evolutionary computation and reinforcement learning. Future work could explore domain-specific heuristics to tailor guiding vectors or employ multi-agent reinforcement learning to cooperatively select diverse variation operators, further enhancing convergence speed and solution quality. Such developments could expand SMAF's practical impact across applications requiring sparse, high-quality solutions.

CRediT authorship contribution statement

ShuZhao Pang: Methodology, Investigation, Software, Writing – original draft, Validation, Visualization, Data curation, Formal analysis. Qingke Zhang: Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing, Software, Supervision, Project administration, Funding acquisition. Zhi-Hui Zhan: Writing – review & editing, Resources. Junjing Li: Writing – review & editing, Resources. Huaxiang Zhang: Writing – review & editing, Resources, Funding acquisition.

Data availability

Data will be made available on request.

Declaration of competing 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.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62006144) and the Taishan Scholar Project of Shandong, China (No. ts20190924).

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms formining association rules. In *20Th VLDB conference* (pp. 487–499).
Bosman, P. A. N., & Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *7*(2), 174–188.
Chu, S.-C., Zhuang, Z., Pan, J.-S., Mohamed, A. W., & Hu, C.-C. (2024). Enhanced sparseEA for large-scale multi-objective feature selection problems. *Complex & Intelligent Systems*, *10*(1), 485–507.
Coppens, R., Reijnen, R., Zhang, Y., Bliek, L., & Steenhuisen, B. (2022). Learning adaptive evolutionary computation for solving multi-objective optimization problems. arXiv:2211.09719.
Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3–18.
Ding, Z., Cao, L., Chen, L., Sun, D., Zhang, X., & Tao, Z. (2023). Large-scale multimodal multiobjective evolutionary optimization based on hybrid hierarchical clustering. *Knowledge-Based Systems*, *266*, 110398.
Ding, Z., Chen, L., Sun, D., & Zhang, X. (2022). A multi-stage knowledge-guided evolutionary algorithm for large-scale sparse multi-objective optimization problems. *Swarm and Evolutionary Computation*, *73*, 101119.
Ding, Z., Chen, L., Sun, D., Zhang, X., & Liu, W. (2024). Efficient sparse large-scale multiobjective optimization based on cross-scale knowledge fusion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*,*54*(11), 6989–7001.
Fischer, A., & Igel, C. (2012). An introduction to restricted boltzmann machines. In *Iberoamerican congress on pattern recognition* (pp. 14–36). Springer.
Geng, H., Shen, J., Zhou, Z., & Xu, K. (2023). An improved large-scale sparse multiobjective evolutionary algorithm using unsupervised neural network. *Applied Intelligence*, *53*(9), 10290–10309.
Hasanzadeh, R., Mojaver, P., Azdast, T., Khalilarya, S., & Chitsaz, A. (2023). Developing gasification process of polyethylene waste by utilization of response surface methodology as a machine learning technique and multi-objective optimizer approach. *International Journal of Hydrogen Energy*, *48*(15), 5873–5886.
He, C., Cheng, R., Zhang, C., Tian, Y., Chen, Q., & Yao, X. (2020). Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers. *IEEE Transactions on Evolutionary Computation*, *24*(5), 868–881.
He, R., Bian, R., Hua, J., Zhao, L., Xu, F., & Long, J. (2025). Multi-objective optimization of gasoline blending scheduling via NSGA-II algorithm with composite operators considering oil movement path planning. *Expert Systems with Applications*, *280*, 127426.
Hu, Z., Nie, X., Sun, H., Wei, L., Zhang, J., & Wang, C. (2025). Sparse large-scale multiobjective optimization algorithm based on impact factor assistance. *Engineering Applications of Artificial Intelligence*, *151*, 110615.
Karimi, F., Dowlatshahi, M. B., & Hashemi, A. (2023). SemiACO: A semi-supervised feature selection based on ant colony optimization. *Expert Systems with Applications*, *214*, 119130.
Khishe, M., Orouji, N., & Mosavi, M. R. (2023). Multi-objective chimp optimizer: An innovative algorithm for multi-objective problems. *Expert Systems with Applications*, *211*, 118734.
Kropp, I., Nejadhashemi, A. P., & Deb, K. (2023). Improved evolutionary operators for sparse large-scale multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, *28*(2), 460–473.
Lalou, M., Tahraoui, M. A., & Kheddouci, H. (2018). The critical node detection problem in networks: A survey. *Computer Science Review*, *28*, 92–117.
Li, B., Zhang, Y., Yang, P., Yao, X., & Zhou, A. (2023). A two-population algorithm for large-scale multi-objective optimization based on fitness-aware operator and adaptive environmental selection. *IEEE Transactions on Evolutionary Computation*, *29*(3), 631–645.
Li, F., Liu, L., & Yu, Y. (2025). Deep reinforcement learning-based multi-objective optimization for electricity–gas–heat integrated energy systems. *Expert Systems with Applications, 262*, 125558.
Li, Y., Feng, X., & Yu, H. (2024). A dynamic knowledge-guided coevolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*,*54*(11), 7054–7064.
Liang, J., Gong, M., Li, H., Yue, C., & Qu, B. (2018). Problem definitions and evaluation criteria for the cec special session on evolutionary algorithms for sparse optimization. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, Report# 2018001.
Liu, J., & Liu, R. (2024). Objective contribution decomposition method and multi-population optimization strategy for large-scale multi-objective optimization problems. *Information Sciences*, *678*, 120950.

Liu, J., Sarker, R., Elsayed, S., Essam, D., & Siswanto, N. (2024). Large-scale evolutionary optimization: A review and comparative study. *Swarm and Evolutionary Computation*, *85*, 101466.

Ma, H., Zhang, Y., Sun, S., Liu, T., & Shan, Y. (2023). A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artificial Intelligence Review*, *56*(12), 15217–15270.

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics : A Journal of Statistics for the Physical, Chemical, and Engineering Sciences*, *42*(1), 55–61.

Ming, F., Gong, W., Wang, L., & Jin, Y. (2024). Constrained multi-objective optimization with deep reinforcement learning assisted operator selection. *IEEE/CAA Journal of Automatica Sinica*, *11*(4), 919–931.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv:1312.5602.

Peng, Y., & Ishibuchi, H. (2020). A decomposition-based large-scale multi-modal multi-objective optimization algorithm. In *2020 IEEE Congress on evolutionary computation (CEC)* (pp. 1–8). IEEE.

Raju, M. R., & Mothku, S. K. (2023). Delay and energy aware task scheduling mechanism for fog-enabled iot applications: A reinforcement learning approach. *Computer Networks*, *224*, 109603.

Ren, J., Qiu, F., & Hu, H. (2023). Multiple sparse detection-based evolutionary algorithm for large-scale sparse multiobjective optimization problems. *Complex & Intelligent Systems*, *9*(4), 4369–4388.

Shang, Q., Tan, M., Hu, R., Huang, Y., Qian, B., & Feng, L. (2025). A multi-stage competitive swarm optimization algorithm for solving large-scale multi-objective optimization problems. *Expert Systems with Applications*, *260*, 125411.

Shao, S., Tian, Y., & Zhang, X. (2024). Deep reinforcement learning assisted automated guiding vector selection for large-scale sparse multi-objective optimization. *Swarm and Evolutionary Computation*, *88*, 101606.

Sharma, S., & Kumar, V. (2022). A comprehensive review on multi-objective optimization techniques: Past, present and future. *Archives of Computational Methods in Engineering*, *29*(7), 5605–5633.

Siraskar, R., Kumar, S., Patil, S., Bongale, A., & Kotecha, K. (2023). Reinforcement learning for predictive maintenance: A systematic technical review. *Artificial Intelligence Review*, *56*(11), 12885–12947.

Sun, Y., & Jiang, D. (2024). An improved problem transformation algorithm for large-scale multi-objective optimization. *Swarm and Evolutionary Computation*, *89*, 101622.

Tian, M., Chen, M., Du, W., Tang, Y., & Jin, Y. (2024a). An enhanced differential grouping method for large-scale overlapping problems. *IEEE Transactions on Evolutionary Computation*.

Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, *12*(4), 73–87. AGSEA implementation available at: https://github.com/BIMK/PlatEMO/tree/master/PlatEMO/Algorithms/Multi-objective%20optimization/AGSEA.

Tian, Y., Feng, Y., Zhang, X., & Sun, C. (2022a). A fast clustering based evolutionary algorithm for super-large-scale sparse multi-objective optimization. *IEEE/CAA Journal of Automatica Sinica*, *10*(4), 1048–1063.

Tian, Y., Li, X., Ma, H., Zhang, X., Tan, K. C., & Jin, Y. (2022b). Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *7*(4), 1051–1064.

Tian, Y., Liu, R., Zhang, X., Ma, H., Tan, K. C., & Jin, Y. (2020a). A multipopulation evolutionary algorithm for solving large-scale multimodal multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, *25*(3), 405–418.

Tian, Y., Lu, C., Zhang, X., Tan, K. C., & Jin, Y. (2020b). Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks. *IEEE Transactions on Cybernetics*, *51*(6), 3115–3128.

Tian, Y., Shao, S., Xie, G., & Zhang, X. (2024b). A multi-granularity clustering based evolutionary algorithm for large-scale sparse multi-objective optimization. *Swarm and Evolutionary Computation*, *84*, 101453.

Tian, Y., Si, L., Zhang, X., Cheng, R., He, C., Tan, K. C., & Jin, Y. (2021). Evolutionary large-scale multi-objective optimization: A survey. *ACM Computing Surveys (CSUR)*, *54*(8), 1–34.

Tian, Y., Zhang, X., Wang, C., & Jin, Y. (2019). An evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, *24*(2), 380–393.

Tran, H.-L., Doan, L., Luong, N. H., & Binh, H. T. T. (2023). A two-stage multi-objective evolutionary reinforcement learning framework for continuous robot control. In *Proceedings of the genetic and evolutionary computation conference* (pp. 577–585).

Verma, S., Pant, M., & Snasel, V. (2021). A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. *IEEE Access*, *9*, 57757–57791.

Wang, Q., Zhang, Q., Meng, F., & Li, B. (2023a). Objective-hierarchy based large-scale evolutionary algorithm for improving joint sparsity-compression of neural network. *Information Sciences*, *640*, 119095.

Wang, X., Zhang, K., Wang, J., & Jin, Y. (2021). An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, *26*(5), 859–871.

Wang, Z., Pei, Y., & Li, J. (2023b). A survey on search strategy of evolutionary multi-objective optimization algorithms. *Applied Sciences*, *13*(7), 4643.

Wang, Z.-J., Zhan, Z.-H., Kwong, S., Jin, H., & Zhang, J. (2020). Adaptive granularity learning distributed particle swarm optimization for large-scale optimization. *IEEE Transactions on Cybernetics*, *51*(3), 1175–1188.

Wang, Z.-J., Zhan, Z.-H., Lin, Y., Yu, W.-J., Wang, H., Kwong, S., & Zhang, J. (2019). Automatic niching differential evolution with contour prediction approach for multimodal optimization problems. *IEEE Transactions on Evolutionary Computation*, *24*(1), 114–128.

Wu, Y., Yang, N., Chen, L., Tian, Y., & Tang, Z. (2023). Directed quick search guided evolutionary algorithm for large-scale multi-objective optimization problems. In *Proceedings of the genetic and evolutionary computation conference* (pp. 777–785).

Xu, Y., Xu, C., Zhang, H., Huang, L., Liu, Y., Nojima, Y., & Zeng, X. (2022). A multi-population multi-objective evolutionary algorithm based on the contribution of decision variables to objectives for large-scale multi/many-objective optimization. *IEEE Transactions on Cybernetics*, *53*(11), 6998–7007.

Xue, B., Zhang, M., & Browne, W. N. (2012). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, *43*(6), 1656–1671.

Yang, X., Zou, J., Yang, S., Zheng, J., & Liu, Y. (2021). A fuzzy decision variables framework for large-scale multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, *27*(3), 445–459.

Zhang, W., Wang, S., Li, G., & Zhang, W. (2023a). Cooperative tri-population based evolutionary algorithm for large-scale multi-objective optimization. *Expert Systems with Applications*, *227*, 120290.

Zhang, Y., Tian, Y., & Zhang, X. (2023b). Improved sparse EA for sparse large-scale multi-objective optimization problems. *Complex & Intelligent Systems*, *9*(2), 1127–1142.

Zhang, Y., Wu, C., Tian, Y., & Zhang, X. (2024). A co-evolutionary algorithm based on sparsity clustering for sparse large-scale multi-objective optimization. *Engineering Applications of Artificial Intelligence*, *133*, 108194.

Zhao, F., Zhou, G., & Wang, L. (2023). A cooperative scatter search with reinforcement learning mechanism for the distributed permutation flowshop scheduling problem with sequence-dependent setup times. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(8), 4899–4911.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: Acomparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271.

Zou, Y., Liu, Y., Zou, J., Yang, S., & Zheng, J. (2023). An evolutionary algorithm based on dynamic sparse grouping for sparse large scale multiobjective optimization. *Information Sciences*, *631*, 449–467.