Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Salp Swarm Algorithm Refinement: Three Core Strategies for Optimization Performance Boost

Dahai Li
Jiangxi University of Science and Technology
Zexiong Liu
6720240906@mail.jxust.edu.cn
Jiangxi University of Science and Technology
Zhendong Wang
Jiangxi University of Science and Technology

Research Article

Keywords: Salp swarm algorithm, Pareto-based leader selection, Fast non-dominated sorting, Scale-free network, State-aware directional tabu search

Posted Date: March 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9051587/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Salp Swarm Algorithm Refinement: Three Core Strategies for Optimization Performance Boost

Dahai Li[1,2], Zexiong Liu[1*], Zhendong Wang[1,2]

[1*]School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, 341000, China .
[2]Jiangxi Provincial Key Laboratory of Multidimensional Intelligent Perception and Control, Jiangxi University of Science and Technology, Ganzhou, 341000, China .

*Corresponding author(s). E-mail(s): 6720240906@mail.jxust.edu.cn;
Contributing authors: 9120130107@jxust.edu.cn;
wangzhendong@jxust.edu.cn;

Abstract

To tackle the deficiencies such as frequent premature convergence and relatively slow convergence rate of the Salp Swarm Algorithm (SSA) incurred by its fitness-prioritized leader selection mechanism and inefficient utilization of population information, this paper proposes three core refinement strategies to boost the performance of SSA, and the refined algorithm named PDMSSA. Firstly, we propose the Pareto-guided leader selection strategy to conduct fast non-dominated sorting based on both the solutions' fitness values and their spatial distribution to ensure leaders are both high-quality and widely distributed in the search space. The scale-free network-based multi-source information fusion mechanism is the second core strategy adopted to refine position updating of followers with the joint guidance of their network-selected neighbor and elite reference solutions. At last, a state-aware directional tabu search mechanism is introduced to suppress redundant search behaviors and facilitating targeted escape from stagnation. The results of Friedman and Wilcoxon rank-sum tests on the experiment data of the refined SSA against with the vanilla SSA, four advanced SSA variants, and five state-of-the-art metaheuristic algorithms on the public CEC2017 and CEC2022 benchmark suites, illustrate that the three core strategies are able to boost up the performance of the SSA efficiently, and the refined SSA can achieve competitive convergence accuracy, stability, and robustness among all co-evaluated algorithms.

Keywords: Salp swarm algorithm, Pareto-based leader selection, Fast non-dominated sorting, Scale-free network, State-aware directional tabu search

1 Introduction

With the increasing scale and structural complexity of engineering and scientific computing problems, classical analytical and deterministic optimization methods, including Newton-type methods[1] and gradient-based approaches[2], often fail to produce solutions of satisfactory global quality within a practical computational budget. This limitation becomes particularly evident when dealing with nonlinear, multimodal, black-box, or high-dimensional objective functions. Such methods typically depend on explicit analytical formulations or accurate gradient information, assumptions that are frequently violated in real-world applications. Moreover, under complex landscapes and constraint interactions, their strong local search bias makes them susceptible to premature convergence to local optima. In response to these challenges, the paradigm of intelligent optimization has progressively evolved into general-purpose metaheuristic frameworks, and it has become a major methodological avenue for tackling complex optimization problems where analytical tractability and gradient information are unavailable or unreliable[3][4]. The contradiction between these classical approaches and discontinuous and non-fungible functions have been promoting researchers to combine stochastic techniques with various optimization strategies, and then in turn accelerated the invention and development of lots of versatile Meta-heuristic Algorithms (MAs).

MAs typically model real world physical laws or biological behaviors to search for the optimal solution within each search space. From the perspective of modeling paradigms and search mechanisms, MAs can be broadly categorized into three major classes: Evolutionary Algorithms (EAs), Physics-based Algorithms (PhAs), and Swarm Intelligence (SIs)–based optimization algorithms. EAs form a foundational class of metaheuristics, whose algorithmic frameworks and developmental trajectories in complex engineering optimization have been systematically reviewed in the literature[5]. Driven by the principles of natural selection and genetic inheritance, these EAs explore the solution space through population-based evolutionary operators. Representative classical algorithms in this category include the Genetic Algorithm (GA)[6] and Differential Evolution (DE)[7]. PhAs are derived from natural physical processes and guide the search process through state transition mechanisms governed by physical laws and analogical dynamics. Their modeling mechanisms and taxonomies have been comprehensively summarized in existing surveys[8]. Typical PhAs include Simulated Annealing (SA)[9], the Gravitational Search Algorithm (GSA)[10], and Heat Transfer Search (HTS)[11], etc. More recently developed algorithms, such as the Photon Search Algorithm (PSA)[12] and the Black Hole Optimizer (BHO)[13], try to model attraction forces or energy-flow dynamics of physical entities to conduct global optimization. SIs represent one of the most widely applied categories of metaheuristics and have been extensively used in various engineering and computational domains, including robotic systems, communication networks, energy systems, and unmanned aerial

vehicle control and path planning[14][15], etc. Their theoretical foundations have been systematically elaborated in the literature[16]. A well-known SI algorithm is Particle Swarm Optimization (PSO)[17], which updates candidate solutions through velocity–position interactions within a population. Other notable SI algorithms include Superb Fairy-wren Optimization Algorithm (SFOA)[18], Coati Optimization Algorithm (COA)[19], and Dung beetle optimizer (DBO)[20], as well as Greylag Goose Optimization(GGO)[21] and Whale Optimization Algorithm (WOA)[22], etc. In addition, SI algorithms such as the Chinese Pangolin Optimizer(CPO)[23], Sparrow Search Algorithm (SSA)[24], and African Vultures Optimization Algorithm (AVOA)[25], which simulate specific perception, foraging, and survival behaviors, have also been widely applied to various engineering optimization problems.

The Salp Swarm Algorithm (SSA) was proposed by Mirjalili et al.[26] in 2017. It imitates the foraging behavior of salp swarms. Given its high performance, relatively strong robustness, and ease of implementation, SSA has become one of the most popular optimization algorithms in recent years. Since its introduction, the SSA has played an important role in a wide range of engineering applications, including function optimization, path planning, energy scheduling, and feature selection. With the deepening of research, however, SSA has also been shown to suffer from several inherent shortcomings, such as a tendency to fall into local optima, slow convergence speed, and limited convergence accuracy. In response to these deficiencies, a variety of improved versions of the Salp Swarm Algorithm have been proposed by researchers.

SSA is a highly efficient SI algorithm and possesses great potential for further improvement and refinement. In past decade, researchers have explored a range of enhancement mechanisms to boost the performance of SSA. Zhang et al.[27] proposed the CDESSA algorithm by introducing logistic-map-based chaotic initialization to improve population uniformity and early exploration, while embedding differential evolution operators into SSA to generate candidate solutions and preserve population diversity during the search process. Tawhid et al.[28] proposed the CSSA algorithm by applying chaotic sequences for population initialization and periodic perturbations during iterations, aiming to enhance early diversity and sustain exploration capability in multimodal optimization problems. El-Shorbagy et al.[29] proposed CS-SSA by adopting chaotic mappings for population initialization and incorporating a chaos-driven search with a shrinking region to transition from global exploration to local exploitation during optimization. Wang et al.[30] proposed OPLSSA by integrating pinhole imaging–based learning with orthogonal experimental design for population initialization and early resampling, aiming to improve distribution uniformity and enhance early exploration while reducing randomness effects. Fan et al.[31] proposed GSSA by adopting a chaotic dyad-based learning strategy to enhance population initialization, followed by population partitioning and genetic operators to support exploration and mitigate randomness-induced instability in SSA. Chen et al[32]. proposed the MSNSSA by partitioning the population into multiple subgroups and integrating symbiosis-based follower learning with non-uniform Gaussian mutation to jointly enhance exploration diversity and exploitation efficiency.

Hybridizing the SSA with complementary optimization techniques has become a popular strategy to tackle its inherent performance limitations. By integrating local

search operators, evolutionary strategies, or learning-based components, hybrid SSA variants seek to improve search efficiency and solution quality in complex optimization problems. Yao et al.[33] developed the IHSSAO by integrating the leadership mechanism of SSA into the position update framework of the Aquila Optimizer(AO)[34], enabling cooperative guidance for global exploration while preserving AO's multi-stage hunting strategies for solution refinement. Singh et al.[35] proposed a hybrid the SSA algorithm, namely HSSAPSO. HSSAPSO integrates the PSO's velocity update into the follower updating mechanism of SSA. Mohamed et al.[36] proposed AOASSA by integrating the arithmetic search mechanism[37] with the cooperative swarm interaction of SSA, aiming to enhance global search behavior and population diversity in complex optimization applications. Chauhan et al.[38] proposed SSCA by integrating Sine–Cosine algorithm[39]'s update operators into SSA, where periodic perturbations with diverse step sizes enhance exploration, while the chain-following mechanism preserves coordinated information propagation during the search process. Li et al[40]. proposed a teaching–learning guided salp swarm algorithm (TLSSA), which integrates teaching and learning phases into SSA to enhance convergence speed and global search capability. Lin et al[41]. developed a double mutational salp swarm algorithm (DMSSA) by embedding cuckoo-inspired and adaptive DE-based mutation schemes, enabling more effective information utilization and a better exploration–exploitation trade-off.

Various refined SSA algorithms have been applied to a wide range of practical engineering problems with multiple non-linear constraints and highly complex search spaces, such as image segmentation, feature selection, and path planning, etc. complex constraints and nonlinear search spaces are commonly encountered. These application-oriented studies further highlight the importance of improving the robustness and adaptability of the SSA algorithm. Sulaiman et al.[42] applied an improved SSA to optimize power flow problems with stochastic photovoltaic generation by incorporating constraint-handling and robustness strategies, enabling reliable optimization of power loss, generation cost, and emission-related objectives under renewable uncertainty. Abualigah et al.[43] proposed MPASSA by hybridizing Marine Predators Algorithm(MPA)[44] with SSA, where SSA enhances swarm cooperation and local search, enabling effective multilevel threshold image segmentation in complex multimodal histograms. Zhao et al.[45] developed EHSSA by introducing multiple search strategies to enhance diversity and convergence in multi-objective image segmentation, reducing premature convergence while improving accuracy and stability across benchmark tests and image experiments. Mahajan et al.[46] incorporated Type-II fuzzy logic into SSA for parameter self-adaptation, enabling dynamic regulation of search behavior based on population diversity and convergence characteristics, and improving robustness in engineering optimization problems under uncertainty. Dagal et al.[47] addressed global maximum power point tracking in photovoltaic systems by hybridizing SSA with PSO, where SSA performs global exploration and PSO refines solutions under dynamically varying irradiance conditions. Rustagi et al.[48] proposed a hybrid SSA–GWO framework for medical image feature selection and classification, combining GWO[49]'s leadership-based global exploration with SSA's chain-following

mechanism to enhance local exploitation and convergence stability in diagnostic applications. According to the aforementioned analysis, the reviewed studies are classified and organized in Table 1 based on their improvement strategies and application contexts.

Table 1: Classification of improved Salp Swarm Algorithm (SSA) variants.

Improvement aspect	Reference	Name	Main Improvement Strategy
Enhancing Diversity or Escaping Local Optima	Zhang et al.[27]	CDESSA	Chaotic initialization based on a logistic map is adopted to generate the initial population, and differential evolution operators are embedded into the SSA updating process.
	Tawhid et al.[28]	CSSA	Chaotic sequences are employed to generate the initial population, and periodic chaotic perturbations are introduced during the optimization process.
	El-Shorbagy et al.[29]	CS-SSA	Chaotic mappings are used for population initialization, combined with a chaos-driven search mechanism with a gradually shrinking search region.
	Wang et al.[30]	OPLSSA	Pinhole imaging–based learning combined with orthogonal experimental design is introduced for population initialization and early resampling.
	Fan et al.[31]	GSSA	A chaotic dyad-based learning strategy is employed to enhance population initialization, followed by population partitioning and genetic operators.
	Chen et al.[32]	MSNSSA	The population is divided into multiple subgroups, where symbiosis-based strategies and non-uniform Gaussian mutation are incorporated to enhance exploitation and maintain population diversity.
Hybrid SSA variants	Yao et al.[33]	IHSSAO	The leadership mechanism of SSA is integrated into the position update framework of the Aquila Optimizer through coordinated updating and information sharing.
	Singh et al.[35]	HSSAPSO	Integrates the PSO velocity update into the follower phase of SSA to strengthen local exploitation and mitigate premature convergence.
	Mohamed et al.[36]	AOASSA	The arithmetic search mechanism of AOA is integrated with the population-based cooperation strategy of SSA.
	Chauhan et al.[38]	SSCA	Sine–cosine operators are incorporated to introduce periodic perturbations, combined with the chain-following structure of SSA.
	Li et al.[40]	TLSSA	Embeds teaching and learning phases to guide population evolution, improving information utilization and convergence behavior.
	Lin et al.[41]	DMSSA	Introduces dual mutation mechanisms inspired by cuckoo search and differential evolution to increase population diversity and escape local optima.
Application-oriented improvements	Sulaiman[42]	ISSA	Constraint-handling and robustness strategies are incorporated into SSA for solving the optimal power flow problem.
	Abualigah et al.[43]	MPASSA	Marine Predators Algorithm is hybridized with SSA to combine global exploration and local search mechanisms.
	Zhao et al.[45]	EHSSA	Multiple search strategies are introduced within the SSA framework.
	Mahajan et al.[46]	Type-II fuzzy SSA	Type-II fuzzy logic is incorporated into SSA to enable parameter self-adaptation based on population diversity and convergence characteristics.
	Dagal et al.[47]	SSA–PSO	SSA and particle swarm optimization are hybridized in serial or parallel configurations.
	Rustagi et al.[48]	SSA–GWO	Grey wolf optimizer mechanisms are integrated with SSA for cooperative global exploration and local exploitation.

Despite significant progress in algorithmic improvements and practical applications of SSA , several fundamental issues remain insufficiently addressed in existing studies.

At first, in most current SSA variants, leader selection is still primarily determined by fitness-based ranking. While such strategies emphasize solution quality, they tend to select leaders that are spatially clustered in multimodal search spaces, leading to homogeneous guidance information. As a result, important population characteristics such as spatial diversity and alternative promising search directions are often underutilized, which significantly increases the risk of premature convergence in complex optimization landscapes. Secondly, the follower update mechanism in existing SSA-based approaches largely preserves the unidirectional chain-following structure of the original algorithm. This single-source information propagation pattern restricts followers to track only limited guidance information, causing the population to move collectively along the same search trajectory and accelerating diversity loss when the search direction becomes suboptimal. At last, despite improvements in global exploration ability, most existing SSA variants still lack explicit mechanisms to regulate search behavior based on historical exploration experience. During the iterative process, populations frequently revisit similar search directions, resulting in redundant explorations and limited local exploitation capability, which in turn constrains the final convergence accuracy.

To address the above unresolved issues and further elevate the performance of SSA, this paper proposes three core refinement strategies to boost the performance of SSA, the three novel strategies are: the fast non-dominated sorting–based leader selection mechanism, the scale-free network–based multi-source information fusion evolutionary strategy and the state-aware directional tabu search mechanism. The refined SSA algorithm with the above three core strategies is named PDMSSA. The main contributions of this paper are summarized as follows:

1. Fast non-dominated sorting–based leader selection mechanism (FNSLSM): FNSLSM treats individuals' fitness values and their relative distance as two competing objectives, and then select leaders from the Pareto front. This mechanism tries to guarantee the selected leaders with high solution quality and sufficient spatial diversity at the same time. Compared with the pure fitness value-based leader selection mechanism, the proposed FNSLSM mechanism can alleviate excessive leader aggregation and mitigates premature convergence, thereby enhancing the algorithm's global exploration capability.

2. Scale-Free Network–Based Multi-Source Information Fusion Evolutionary Strategy (SFNMIFS): SFNMIFS adopts a scale-free network–based interaction structure to establish topological connections among individuals. This strategy integrates topology-guided neighborhood information with dual-elite guidance while preserving the original chain-following characteristic, followers are enabled to respond to multiple high-quality information sources. It tries to overcome the limited information propagation and single-source following behavior of followers in the vanilla SSA.

3. State-Aware Directional Tabu Search Mechanism (SADTSM): SADTSM uses a vector-based tabu list to record historically ineffective exploration directions to tackle stagnation and redundant exploration in the later evolution stages of the vanilla SSA algorithm. By evaluating the directional similarity between

current search moves and tabu directions, SADTSM can avoid repetitive and low-efficiency searches effectively, therefore it can enhance local exploitation accuracy while improving the ability to escape local optima.

The remainder of this paper is organized as follows. Section 2 introduces the fundamental principles of the original Salp Swarm Algorithm (SSA). Section 3 provides a detailed analysis and description of the individual improvement strategies incorporated into PDMSSA. Section 4 presents the experimental design and a thorough analysis of the experimental results. Section 5 applies PDMSSA to an unmanned aerial vehicle (UAV) path planning problem. Finally, Section 6 concludes the paper and outlines potential directions for future research and further improvements.

2 Salp Swarm Algorithm

The SSA is a population-based metaheuristic inspired by the swarming behavior of salps, a class of pelagic organisms that move in chain-like formations. SSA abstracts the collective movement and information transmission mechanisms observed in salp chains to guide the search process toward promising regions of the solution space. In SSA, the population is divided into two functional roles: leaders and followers. Leaders are responsible for guiding the overall search direction by exploring the neighborhood of the current food source (i.e., the best solution found so far), while followers update their positions by tracking the individual immediately ahead of them in the chain. This hierarchical interaction enables information propagation from leaders to followers, thereby coordinating global exploration and local exploitation. The salp population is represented as a two-dimensional matrix of size $N \times D$, as shown in Eq. 1, where N denotes the population size and D is the dimensionality of the optimization problem.

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,j} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,j} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,j} & \cdots & x_{i,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,j} & \cdots & x_{N,D} \end{bmatrix} \tag{1}$$

In the vanilla SSA, only the first individual is designated as the leader, and all remaining individuals act as followers. This single-leader architecture causes the population's search direction to rely heavily on a single solution, which increases the risk of premature convergence when the leader is trapped in a local optimum. Moreover, the absence of parallel guidance limits exploration efficiency in complex search landscapes. To overcome this limitation, recent studies adopt a multi-leader strategy commonly by assigning the first $\frac{N}{2}$ individuals as leaders and the remaining individuals as followers [50][51]. This variation enables parallel exploration from multiple promising regions and enhances population diversity. Nowadays, all subsequent proposed algorithmic enhancements for the SSA were developed based on this structural setting.

$$X_{i,j} = \begin{cases} F_j + c_1(ub_j - lb_j)c_2 + lb_j, & c_3 \geq 0.5 \\ F_j - c_1(ub_j - lb_j)c_2 + lb_j, & c_3 < 0.5 \end{cases} \tag{2}$$

The position update of a leader is defined by Eq. 2, where F_j denotes the current food source position in the j-th dimension, ub_j and lb_j represent the upper and lower bounds of the search space in the j-th dimension, and c_2 and c_3 are uniformly distributed random variables in the range of $[0, 1]$, which determine step size of the leader's position update and the direction. The coefficient c_1, defined in Eq. 3, controls the step size of leader movements. Specifically, a larger value of c_1 at early iterations allows leaders to explore a wider search region, while its gradual decrease over iterations restricts movement amplitudes and promotes local exploitation near the food source.

$$c_1 = 2e^{-\left(\frac{4t}{T}\right)^2} \tag{3}$$

$$x_j^t = \frac{1}{2}\left(x_j^t + x_j^{t-1}\right) \tag{4}$$

Followers update their positions using a simplified kinematic model inspired by Newtonian motion. Under the assumptions of zero initial velocity and a unit time interval, the follower update as Eq. 4 is obtained by averaging the current position with that of the preceding individual. This chain-based update mechanism facilitates smooth information propagation along the population and contributes to stable convergence behavior. Since follower updates are indirectly influenced by the food source through the leader–follower chain, rather than being directly attracted to it, a certain level of population diversity can be maintained during the search process. The pseudocode of the vanilla SSA is summarized in Algorithm 1.

3 The proposed PDMSSA

3.1 Fast Non-Dominated Sorting–Based Leader Selection Mechanism (FNSLSM)

In the SSA, leaders occupy the head positions of the population chain, and their primary role is to guide the population towards to the food source while exploring and exploiting high-quality potential regions, therefore, leaders are expected to possess both high fitness values and a distribution that is as uniform as possible across the population. A well-designed leader selection mechanism is essential for enhancing the global search capability of the algorithm and for preventing premature convergence caused by early aggregation of population individuals. The vanilla SSA employs a pure fitness-based ranking mechanism to select leaders. This overly greedy selection strategy exhibits pronounced structural deficiencies when dealing with high-dimensional multimodal objective functions, because this greedy leader selection mechanism relies on fitness values only while completely disregarding the spatial topological information of the population. However, in multimodal landscapes, these individuals with high-fitness values are often highly clustered around the same local optimum basin.

Algorithm 1 Pseudocode of Salp Swarm Algorithm (SSA)

Require: Objective function $f(\mathbf{x})$, bounds $[\mathbf{lb}, \mathbf{ub}]$, population size N, maximum iterations T.

Ensure: Best solution $\mathbf{X}_{best}$ and its global optimal fitness value F_{best}.

Initialize population $\{\mathbf{x}_i^{(0)}\}_{i=1}^{N}$ uniformly in $[\mathbf{lb}, \mathbf{ub}]$;
Evaluate fitness $f(\mathbf{x}_i^{(0)})$ for all $i \in \{1, \ldots, N\}$;
Sort population based on fitness in ascending order; ▷ Ensure top 50% are at the front
$\mathbf{X}_{best} \leftarrow \mathbf{x}_1^{(0)}$; ▷ The first individual is the best after sorting
$F_{best} \leftarrow f(\mathbf{X}_{best})$;
for $t = 1$ to T **do**
    Update parameter c_1 according to Eq. 3;
    / Leader Update Phase (Top 50%) */*
    for $i = 1$ to $N/2$ **do**
        Draw random numbers $c_2, c_3 \sim \mathcal{U}(0, 1)$;
        Update the position of the leader salp $\mathbf{x}_i^{(t)}$ based on Eq. 2;
    end for
    / Follower Update Phase */*
    for $i = N/2 + 1$ to N **do**
        Update the position of the follower salp $\mathbf{x}_i^{(t)}$ based on Eq. 4;
    end for
    / Boundary Control and Evaluation */*
    for $i = 1$ to N **do**
        $\mathbf{x}_i^{(t)} \leftarrow \min(\max(\mathbf{x}_i^{(t)}, \mathbf{lb}), \mathbf{ub})$;
        Evaluate fitness $f(\mathbf{x}_i^{(t)})$;
    end for
    Sort population based on fitness in ascending order;
    Update $\mathbf{X}_{best}$ and F_{best} if a better solution is found;
end for
return $\mathbf{X}_{best}, F_{best}$;

This "fitness-prioritized, distribution-neglected" selection paradigm inevitably leads to two adverse effects. It causes severe homogenization of the leader group in the decision space with high probability, thereby losing the ability to guide the population to spread in diverse directions. On the other hand, individuals located in sparsely distributed regions, whose objective function values may be temporarily inferior but who possess high exploratory potential, are systematically marginalized. Ultimately, the lack of spatial information in leader selection results in insufficient leader diversity, rendering the algorithm highly susceptible to irreversible premature convergence due to the absence of multi-path search support when confronting complex multimodal optimization problems.

To overcome these limitations, we propose the fast non-dominated sorting–based leader selection mechanism, namely FNSLSM. The goal of FNSLSM is to construct a selection strategy that can explicitly balances convergence and diversity by jointly characterizing solution quality and spatial distribution within the population. FNSLSM adopts two complementary observation metrics defined for population individuals: fitness ranking and relative distance.

In single-objective optimization, the fitness value serves as the fundamental indicator of solution quality. For minimization problems, smaller fitness values correspond to superior solutions, and the regions occupied by such individuals can be regarded as high-quality areas of the search space, typically containing local or global optima together with their basins of attraction. For continuously differentiable problems, individuals with better fitness values tend to aggregate progressively along the descent directions during the evolutionary process. Let the population size be N, and denote the fitness value of the j-th individual at iteration t by F_j^t. By sorting all individuals in ascending order of their fitness values, the ranking index of the i-th individual is denoted as p_i^t, where $i \in \{1, 2, \ldots, N\}$. All individual with ranking index $k < i$ exhibits superior fitness compared with the individual ranked at position i. The fitness ranking index therefore provides an ordinal measure for quantifying the relative solution quality of individuals in the current iteration.

While retaining high-ranking individuals is essential for preserving convergence performance, excessive selection pressure can accelerate population aggregation and lead to a rapid loss of diversity. To alleviate this issue, FNSLSM applies an additional diversity-related metric based on the relative distance. For an individual with fitness ranking index p_i^t, the relative distance l_i^t is defined as the minimum Euclidean distance between this individual and the set of individuals with better fitness rankings, i.e., $\left\{p_1^t, p_2^t, \ldots, p_{i-1}^t\right\}$, as formally given in Eq. 5. A small value of l_i^t indicates that the individual is located in close proximity to dominant solutions, implying strong redundancy in spatial and directional information and a limited contribution to population evolution. In contrast, a large value of l_i^t suggests that the individual resides in a relatively unexplored region of the search space that has not yet been covered by higher-quality individuals, even if its fitness ranking is not among the best. For the individual with ranking index P_1^t, the corresponding distance is defined as infinity, reflecting its role as the reference solution in the distance-based diversity evaluation.

$$l_i^t = \min_{1 \le m \le p_i^{t-1}} \left\{ \sqrt{\sum_{j=1}^{D} \left(x_{p_i,j}^t - x_{m,j}^t\right)^2} \right\} \tag{5}$$

After obtaining the two-evaluation metrics, fitness ranking and relative distance, our goal is to pick up leaders with the minimum fitness ranking index p_i^t while the maximum relative distance l_i^t. A feasible method is linear scalarization to construct a composite indicator m_i^t by using a weighted-sum scheme, such as $m_i^t = (1-k)\, p_i^t + k l_i^t$ or its normalized variant. However, linear aggregation–based scalarization suffers from inherent limitations that restrict its effectiveness in practice. At first, its performance is highly sensitive to the weighting coefficient k, whose appropriate value is strongly problem-dependent. While smaller values of kmay accelerate convergence in simple landscapes, they often induce premature convergence in multimodal or highly complex problems, where larger values are required to preserve diversity. The lack of theoretical guidance for weight assignment typically leads to ad hoc tuning, increasing computational cost and reducing reproducibility. Secondly, from a geometric perspective, the weighted-sum method can only identify solutions on the convex regions of the Pareto front and is incapable of capturing non-convex trade-offs. Consequently, individuals that exhibit large relative distances but temporarily inferior fitness, often crucial for

escaping local optima, are prone to being discarded. Thirdly, linear coupling of heterogeneous indicators amplifies the impact of noise in fitness evaluation, leading to instability in the composite ranking and frequent fluctuations in leader selection, which degrades convergence stability in later search stages.

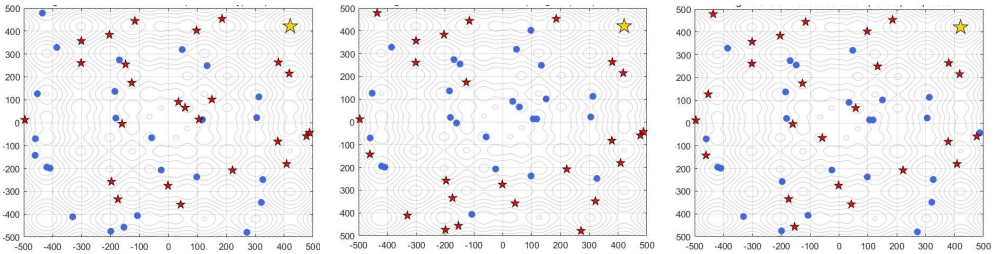
To overcome these limitations, FNSLSM adopts the fast non-dominated sorting[52] to conduct the leader selection. Instead of relying on greedy fitness-based selection or scalarized indicators, FNSLSM formulates the leader selection as a bi-objective minimization problem with respect to the fitness ranking and the relative distance. In FNSLSM, a bi-objective vector is defined for each individual in the population as $F_i = [p_i, -l_i]$, where p_i denotes the ranked fitness value reflecting convergence performance, and l_i represents the relative Euclidean distance between individual i and the sorted population, which quantifies its contribution to diversity. By negating l_i, the original diversity maximization objective is converted into a minimization form, ensuring consistency in Pareto dominance evaluation across objectives. Based on the above formulation, FNSLSM partitions the whole population into a set of non-dominated fronts $\{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_k\}$. Individuals in the first front $\mathbf{F}_1$ achieve Pareto optimality in terms of convergence and diversity, indicating that no other solutions can simultaneously improve fitness without sacrificing distribution. Lower-ranked fronts represent increasingly dominated trade-offs. Leader selection is then performed by sequentially incorporating individuals from $\mathbf{F}_1$ to $\mathbf{F}_k$ until the predefined leader size N_{leader} is reached. When the inclusion of an entire front $\mathbf{F}_k$ exceeds this limit, a secondary selection criterion based on the ideal point distance is introduced instead of random truncation. For individuals in the critical front $\mathbf{F}_k$, the normalized Euclidean distance to the ideal point $z^* = [\min(p_i), \min(-l_i)]$. in the objective space is computed, and those closest to $\mathbf{z}^*$ are preferentially retained. This refinement not only preserves the Pareto-optimal characteristics of the leader set but also ensures a smooth transition from exploration to exploitation, effectively mitigating the loss of high-quality solutions caused by random truncation.

FNSLSM directly identifies non-inferior solutions based on Pareto dominance relationships, thereby ensuring diversity in the distribution of the leader set within the objective space. This enables the algorithm to explore multiple promising regions in parallel and, at a mechanistic level, supports strong global search capability in complex multimodal environments. In addition, non-dominated sorting eliminates sensitivity to the selection of weighting coefficients and the need for objective normalization, reducing the risk of erroneously discarding solutions that exhibit outstanding performance in a single objective but are undervalued under weighted aggregation. Furthermore, because this selection framework is driven by objective vectors rather than a specific scalar evaluation function, it possesses inherent structural extensibility. It can naturally accommodate constraint-handling mechanisms or the incorporation of additional evaluation criteria, thereby providing a unified interface for the continued evolution and enhancement of the algorithm.

FNSLSM directly identifies non-inferior solutions based on Pareto dominance relationships, thereby ensuring diversity in the distribution of the leader set within the objective space. This enables the algorithm to explore multiple promising regions in parallel and, at a mechanistic level, supports strong global search capability in complex

multimodal environments. In addition, non-dominated sorting eliminates sensitivity to the selection of weighting coefficients and the need for objective normalization, reducing the risk of erroneously discarding solutions that exhibit outstanding performance in a single objective but are undervalued under weighted aggregation. Furthermore, because this selection framework is driven by objective vectors rather than a specific scalar evaluation function, it possesses inherent structural extensibility. It can naturally accommodate constraint-handling mechanisms or the incorporation of additional evaluation criteria, thereby providing a unified interface for the continued evolution and enhancement of the algorithm.

To analyze the influence of different leader selection strategies, Fig. 1 illustrates the leaders picked by three leader selection strategies on the same objective function, respectively. In the figure, red pentagrams denote the selected leaders, while the remaining individuals act as followers. As shown in Fig. 1a, the fitness value-only strategy tends to concentrate leaders within localized regions, leading to limited coverage of the search space. As shown in Fig. 1b, incorporating a weighted combination of fitness and relative distance can partially alleviates this issue by expanding leader distribution toward boundary regions, although clustering and overlap remain in some areas. In contrast, the fast non-dominated sorting–based bi-objective strategy yields a more balanced distribution across the search space, with leaders covering diverse regions, as illustrated in Fig. 1c. This fact indicates that the proposed strategy achieves a more effective trade-off between convergence and population diversity, thereby providing more stable support for global exploration in complex search spaces.



(a) Producers picked by the vanilla SSA

(b) Producers picked by the fitness-relative distance weighted

(c) Producers picked by the FNSLSM

Fig. 1: Comparison of population distributions under different leader selection strategies.

3.2 Scale-Free Network–Based Multi-Source Information Fusion Evolutionary Strategy (SFNMIFS)

As indicated in Eq. 4, the follower's position update mechanism in the vanilla SSA is inspired by the chain-like cooperative behavior observed in biological swarms. However, the follower's position update imposes inherent limitations on information interaction, which restrict the algorithm's performance in complex and high-dimensional search spaces. Firstly, the vanilla SSA employs a strictly unidirectional linear chain topology, where information exchange occurs only between adjacent individuals. Under this structure, global search information propagates slowly from the leader to individuals at the tail of the chain, resulting in delayed population response in dynamic or multimodal environments. Moreover, this serial dependency makes the information flow vulnerable to local stagnation: once an upstream individual is trapped in a local optimum, misleading information will be repeatedly transmitted along the chain, increasing the risk of premature convergence at the population level. At the second, the follower update process relies primarily on linear combinations of neighboring positions, which essentially acts as a local smoothing operation. Such updates exploit only limited geometric information and fail to incorporate guidance from other high-quality individuals distributed across the population. In multimodal optimization problems, this locally confined information usage often leads to ambiguous search directions and restricts the population's ability to effectively explore promising regions of the search space.

These limitations imply the fact that improving the search capability of SSA followers requires escaping the single chain-based dependency structure and adopting a more expressive information interaction paradigm. An effective mechanism should preserve local cooperation while enabling individuals to integrate guidance from multiple high-quality information sources. From this perspective, we propose the SFNMIFS to provides a natural and well-motivated means of enhancing information dissemination and utilization within the population. SFNMIFS tries to construct a population interaction characteristic by mapping the whole population to hub nodes of a scale-free network, and leverages the short average path length and heterogeneous connectivity of hub nodes to promote efficient dissemination of high-quality information across the population. Based on this scale-free network, SFNMIFS reformulates the position update rule of followers by incorporating two complementary components: an active learning term derived from the individual's network neighborhood and a cooperative guidance term jointly formed by the global best solution and contemporary elite individuals. This modification transforms the update process from a single chain-dependent following mode into a multi-source information–driven cooperative mechanism, enabling followers to exploit both local neighborhood information and global guidance.

SFNMIFS tries to restructure the interaction topology among individuals in the population by using a scale-free network. A scale-free networks is a specific type of network architecture in which the degree distribution of nodes (i.e., the number of edges incident to each node) adheres a power-law distribution, namely the degree k of each node satisfies $P(k) \sim k^{-\gamma}$ with $2 < \gamma < 3$. This distribution property gives rise to significant topological heterogeneity in the network's topology, endowing it with

powerful information-propagation capabilities and robustness. Scale-free networks are usually distinguished by a small number of nodes emerge as highly connected hubs, known as "hub" nodes. Such heterogeneity leads to short average path lengths, which facilitates efficient dissemination of high-quality information across the population.

At first, SFNMIFS applies the Barabási–Albert (BA) model[53] to constructs a scale-free network by following steps: The network is initiated from a fully-connected undirected subgraph containing 5 nodes. Then, all individuals are mapped as newly added network nodes one by one based on their ranking order after the fast non-dominated sorting conducted by FNSLSM described in Section 3.1, such that "better" individuals with higher Pareto ranks are mapped to highly connected "hub" nodes, while lower-ranked individuals occupy peripheral positions. Secondly, based on the constructed network, all individuals can explore their one-hop and two-hop neighborhoods by conducting breadth-first search on the constructed network and selects the neighbor with the best fitness value as its network-guided reference as shown in Fig. 2. Therefore, the followers of SSA, which are mapped as the peripheral nodes of the network, can access informative guidance beyond their local neighborhoods by their best "neighbors" searched from the network.

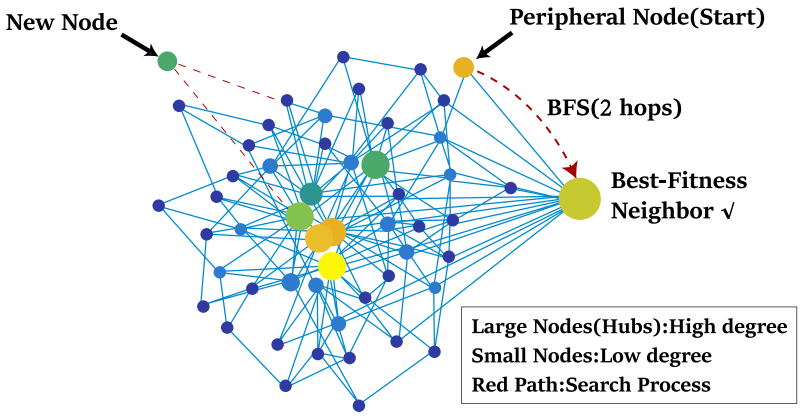


Fig. 2: Illustration of the scale-free network topology and BFS-based neighbor selection mechanism.

In addition, SFNMIFS adopts a dual-elite cooperative guidance term to further refine the position update of followers. This term jointly incorporates the historical global best position G_{best} and the current population best position P^t_{best}. Specifically, G_{best} serves as a long-term memory of the evolutionary process, providing stable global guidance and maintaining the population's overall convergence tendency toward promising regions. In contrast, P^t_{best} reflects the instantaneous optimal state of the population at iteration t, capturing recent search dynamics and local distribution characteristics. In multimodal optimization scenarios, where the historical best solution may become trapped in local optima, the real-time feedback provided by P^t_{best} enables

adaptive correction of follower search directions. The integration of these two complementary information sources enhances directional awareness while improving the robustness of the update process.

At last, SFNMIFS refines the followers' position update by taking their best "neighbors" $\mathbf{X}^t_{\text{neighbor}}$ and the dual-elite cooperative guidance term into consideration. The refined update equation is given in Eq. 6, where r_1 and r_2 are independent random variables uniformly distributed in $[0, 1]$, $\mathbf{X}^t_{\text{neighbor}}$ denotes the position of the best neighboring individual determined by the scale-free network, G_{best} represents the historical global best position, and P^t_{best} denotes the population best position at iteration t.

$$X_i^{t+1} = \frac{X_i^t + X_i^{t-1}}{2} + r_1 \left(X^t_{\text{neighbor}} - X_i^t\right) + r_2 \left(\frac{G_{\text{best}} + P^t_{\text{best}}}{2} - X_i^t\right) \tag{6}$$

As illustrated in Fig. 3, SFNMIFS reconstructs the follower update pattern from the single, passive chain-following behavior in the vanilla SSA to a multi-source cooperative evolutionary process jointly driven by chain-based inertia preservation, scale-free network neighborhood interactions, and dual-elite guidance. By introducing lateral neighborhood information and longitudinal elite guidance while retaining the original structural stability, this mechanism enhances followers' responsiveness to effective search directions on the basis of maintaining global exploration breadth. Consequently, the algorithm achieves improved convergence accuracy and enhanced robustness during the local search phase when addressing complex multimodal optimization problems.

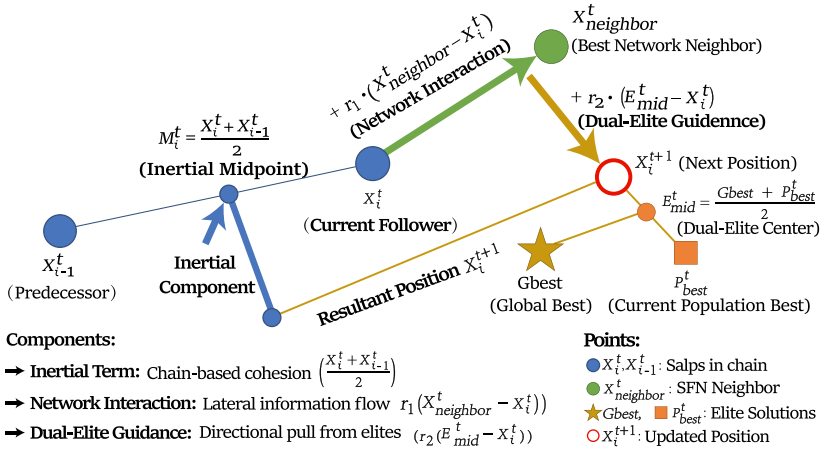


Fig. 3: Illustration of the multi-source information fusion and cooperative guidance mechanism for follower position updating.

3.3 State-Aware Directional Tabu Search Mechanism (SADTSM)

In the vanilla SSA, leaders govern the trajectory of population exploration and exploitation. However, when navigating high-dimensional complex landscapes, the leader position update mechanism exhibits three structural deficiencies: lack of scale adaptability, directional blindness, and absence of state awareness. Firstly, the update step size is linearly coupled with the search space boundaries defined by $ub_j - lb_j$. In problems involving significant decision variable magnitudes or expansive search domains, this rigid step size lacks an adaptive contraction mechanism, easily inducing severe position oscillations and hampering fine-grained search within the vicinity of local optima. Second, the directional sampling of leaders is essentially a memoryless stochastic process driven by coefficients c_2 and c_3. Due to the failure to utilize historical search feedback, leaders often conduct redundant exploration along vectors already proven ineffective or inefficient. Finally, the standard SSA lacks an intrinsic mechanism for monitoring evolutionary states. When the algorithm falls into a "stagnation-oscillation" loop, the system cannot perceive such pathological states to proactively reconfigure its search strategy. Furthermore, followers, constrained by the rigid chain-based topology, amplify the population-level blindness caused by leader decision errors through a cascading effect. Consequently, leader failure inevitably drives the entire population into evolutionary stagnation.

To address these deficiencies, this paper proposes the State-Aware Directional Tabu Search Mechanism (SADTSM). Leveraging the core advantage of Tabu Search[54][55]—dynamic memory structures—SADTSM mitigates the memoryless limitations of SSA by systematically evading inefficient paths and adaptively activating auxiliary strategies based on population status. The SADTSM integrates three synergistic components: the State-Aware Triggering Mechanism (SATM), Hybrid Neighborhood Solution Generation (HNSG), and Directional Tabu with Dynamic Tabu Tenure (DT-DTT). Firstly, SATM functions as the system's "gatekeeper" monitoring the evolutionary status in real-time to precisely determine the intervention timing of auxiliary strategies upon detecting stagnation or inefficient oscillation. Subsequently, HNSG constructs a set of high-quality candidate solutions with complementary attributes, enhancing the ability to escape local optima while simultaneously intensifying the exploitation of promising regions. Finally, DT-DTT dynamically adjusts the tabu tenure to suppress repeated exploration along directions yielding inefficient gains, thereby guiding the search trajectory toward global optimality.

3.3.1 State-Aware Triggering Mechanism (SATM)

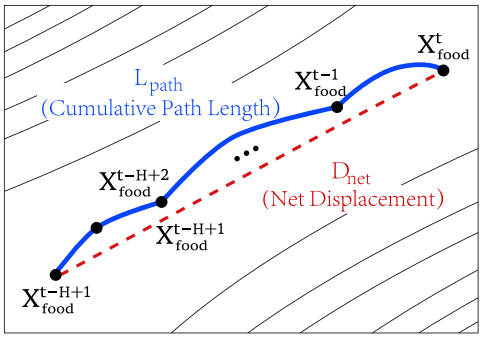
To establish a dynamic equilibrium between escaping local optima and ensuring computational efficiency, the SADTSM integrates a trajectory-fitness coupled triggering mechanism, namely the SATM. Distinct from the rigid fixed-frequency activation protocols prevalent in traditional hybrid methods, this mechanism activates the Tabu Search (TS) module in an on-demand fashion, initiating intervention only when pathological evolutionary states (e.g., inefficient oscillation or evolutionary stagnation) are detected. Specifically, a sliding window of length H is maintained to record

the historical trajectory snapshots of the global best solution $\mathbf{X}_{food}$, denoted as $\mathbf{T} = \{\mathbf{X}_{\text{food}}^{t-H+1}, \ldots, \mathbf{X}_{\text{food}}^{t}\}$. Correspondingly, a Trajectory Index (TI) is introduced to quantify the geometric linearity of the search path (Eq. 7), where D_{net} represents the Euclidean net displacement between the start and end points of the sliding window, and L_{path} denotes the accumulated path length across generations. When $TI \to 1$, the trajectory exhibits quasi-linear characteristics, indicating that the population is performing an efficient ballistic search toward potential global optima; thus, no external intervention is required as shown in Fig. 4a. In contrast, as $TI \to 0$, the trajectory becomes highly tortuous, which implies that the search behavior has degenerated into disordered Brownian motion within a confined region, this phenomenon signifies that the algorithm has been trapped in the attraction basin of a local optimum as illustrated in Fig. 4b

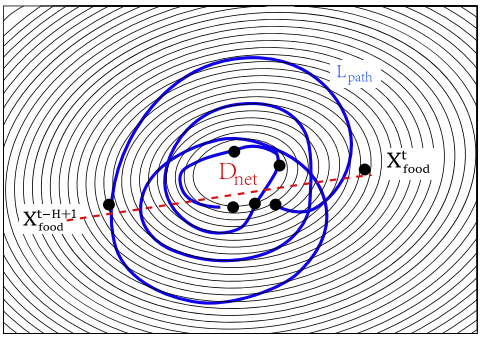
$$TI = \frac{D_{\text{net}}}{L_{\text{path}}} = \frac{\left\| X_{\text{food}}^{t} - X_{\text{food}}^{t-H+1} \right\|_2}{\sum_{k=0}^{H-1} \left\| X_{\text{food}}^{t-k} - X_{\text{food}}^{t-k-1} \right\|_2} \tag{7}$$

$$\Delta f = \left| f\left(X_{\text{food}}^{t}\right) - f\left(X_{\text{food}}^{t-1}\right) \right| < \epsilon_f \tag{8}$$

To preclude false positives arising solely from positional data, a fitness stagnation criterion is further introduced as an auxiliary constraint. As defined in Eq. 8, fitness stagnation is identified when the improvement Δf between consecutive iterations drops below a precision threshold $\boldsymbol{\epsilon_f}$. Notably, the triggering mechanism identifies algorithmic failure and activates the TS module if and only if both spatial tortuosity (TI ≈ 0) and fitness stagnation ($\boldsymbol{\Delta f} < \boldsymbol{\epsilon_f}$) are simultaneously satisfied. To prevent excessive intervention from disrupting the intrinsic evolutionary inertia of the population, a dynamic cooling window $\mathbf{T_{cool}}$ is additionally implemented. Subsequent to each TS module activation , the algorithm is required to undergo a standard evolutionary phase for $\mathbf{T_{cool}}$ generations , which ensures a stable temporal balance between local optima escape maneuvers and the global convergence process.



(a) near-linear evolution with high directional consistency.



(b) highly tortuous trajectory indicating local oscillation.

Fig. 4: Illustration of the trajectory-curvature-based state-aware triggering mechanism.

3.3.2 Hybrid Neighborhood Solution Generation (HNSG)

To establish a reference set that effectively balances quality excavation and spatial coverage, the HNSG selects the top 20% elite individuals ranked by their fitness values from the current population, supplemented by a 10% random subset sampled from the remaining non-elite individuals. Subsequently, for each reference solution, two distinct candidate neighbors are generated via complementary strategies to facilitate competitive selection.

The first candidate, denoted as X_{cand1}, is synthesized using a momentum-enhanced differential perturbation strategy designed to reinforce the precision and coherence of local exploitation. Drawing inspiration from the momentum method in gradient descent, a gradient-free momentum mechanism is adopted to smooth stochastic perturbations by using historical search inertia. Specifically, X_{cand1} is constructed as a linear combination of three synergistic components, as shown in Eq. 9: a population-difference term $\mathbf{\Delta}_{\text{diff}}$ that utilizes inter-individual differences to maintain adaptive diversity; an elite-guided term $\mathbf{\Delta}_{\text{elite}}$ that integrates dual guidance from the global best X_{Gbest} and current best X^t_{Pbest} to provide strong convergence traction; and a historical momentum term (V^t_{mom}). To ensure stability, the momentum vector is updated via an Exponential Moving Average (EMA) scheme as shown in Eq. 10, where X^t_{best} denotes the superior candidate survivor selected after the greedy tabu evaluation, and X^t_{base} represents the corresponding generating base solution. The parameter λ acts as a memory decay factor, comprehensively utilizing both the accumulated historical directional memory and the latest effective step information.

In contrast, the second candidate X_{cand2} is generated via fitness-weighted quasi-opposition learning, addressing the inherent limitations of traditional Opposition-Based Learning (OBL). Conventional OBL typically employs the geometric center of the population as a mirror reference, treating all individuals as "equal-mass particles" and neglecting fitness heterogeneity. This often results in "blind opposition," where generated solutions deviate from high-value regions. To overcome this, a Fitness-Weighted Centroid $\mathbf{X}_{\text{wc}}$ is introduced (Eq. 11), which maps fitness values to physical "mass" thereby causing the centroid to adaptively shift toward regions with a high concentration of superior solutions. Based on this centroid, a Quasi-Opposition mechanism is employed to generate X_{cand2}. Instead of performing a full opposition (Eq. 12) jump which risks overshooting, the candidate is uniformly sampled within the interval bounded by $\mathbf{X}_{\text{wc}}$ and the generalized opposite point X_{opp} (Eq. 13). This strategy retains the capability for large-scale exploration while significantly enhancing the probability of fine-grained detection near the weighted centroid.

$$X_{\text{cand1}} = X_{\text{base}} + \mathbf{\Delta}_{\text{diff}} + \mathbf{\Delta}_{\text{elite}} + \omega \cdot V^t_{\text{mom}} \tag{9}$$

$$V^{t+1}_{\text{mom}} = \lambda \cdot V^t_{\text{mom}} + (1-\lambda) \cdot \left(X^t_{\text{best}} - X^t_{\text{base}}\right) \tag{10}$$

$$w_i = \frac{f_{\text{worst}} - f_i}{\sum_{j=1}^{N} (f_{\text{worst}} - f_j)}, \quad X_{\text{wc}} = \sum_{i=1}^{N} w_i \cdot X_i \tag{11}$$

$$X_{\text{opp}} = 2X_{\text{wc}} - X_{\text{base}} \tag{12}$$

Finally, according to the greedy fitness-based criterion defined in Eq. 14, the optimal solution is selected from the generated candidate solutions as the final neighborhood solution for each initial solution.

$$X_{\text{cand2},i,j} = \begin{cases} \text{rand}(\min(X_{\text{wc},j},\, X_{\text{opp},i,j}),\max(X_{\text{wc},j},\, X_{\text{opp},i,j})), & r_j < 0.5, \\ X_{\text{base},i,j}, & \text{otherwise.} \end{cases} \tag{13}$$

$$X_{\text{candidate}} = \arg \min_{X \in \{X_{\text{cand1}},\, X_{\text{cand2}}\}} f(X) \tag{14}$$

3.3.3 Directional Tabu with Dynamic Tabu Tenure (DT-DTT)

To mitigate the directional aimlessness and repetitive path exploration inherent in standard SSA, SADTSM integrates the DT-DTT mechanism to tackle deficiencies primarily stemming from the leader's reliance on memoryless stochastic perturbations. Unlike conventional tabu strategies that explicitly prohibit revisiting specific coordinate positions, the proposed DT-DTT establishes a vectorial memory to record ineffective exploration directions that have empirically yielded marginal or negligible fitness gains. By proscribing redundant exploration along these non-productive vectors, the algorithm effectively curtails oscillatory trajectories within local regions and eliminates inefficient backtracking behaviors.

$$\max_{t_k \in TL} (d_{\text{move}} \cdot t_k) > \theta_{\cos} \tag{15}$$

DT-DTT maintains a dynamic tabu list $TL = \{\widehat{t_1}, \ldots, \widehat{t_L}\}$, in which each element $\widehat{t_l}$ denotes a normalized direction vector corresponding to a historically unsuccessful search move. For a candidate displacement from a base solution X_{base} to a neighbor $X_{\text{candidate}}$, the displacement vector is defined as $v_{\text{move}} = X_{\text{candidate}} - X_{\text{base}}$, and the corresponding unit exploration direction is given by $\widehat{d_{\text{move}}} = \frac{v_{\text{move}}}{\|v_{\text{move}}\|_2}$. DT-DTT determines the tabu status determined by evaluating the geometric similarity between the current direction and those stored in $\mathcal{TL}$. If the cosine similarity between $\widehat{d_{\text{move}}}$ and any recorded vector exceeds a predefined threshold $\theta_{\cos}$, the move is classified as directionally consistent with known ineffective behaviors and is consequently prohibited. The criterion is expressed as Eq. 15.

From a geometric perspective, the directional tabu criterion can be interpreted as constructing a conical exclusion region in the direction space for each inefficient direction stored in the tabu list, where the historical direction vector acting as the cone axis as shown in Fig. 5. Specifically, search trajectories falling within these cones are deemed highly correlated with previous ineffective exploration attempts and thus are blocked to avoid search blindness. Conversely, exploration directions outside these exclusion regions retain their novelty and are permitted for subsequent search steps, thereby ensuring the algorithm maintains effective exploratory capability while avoiding redundant failures.

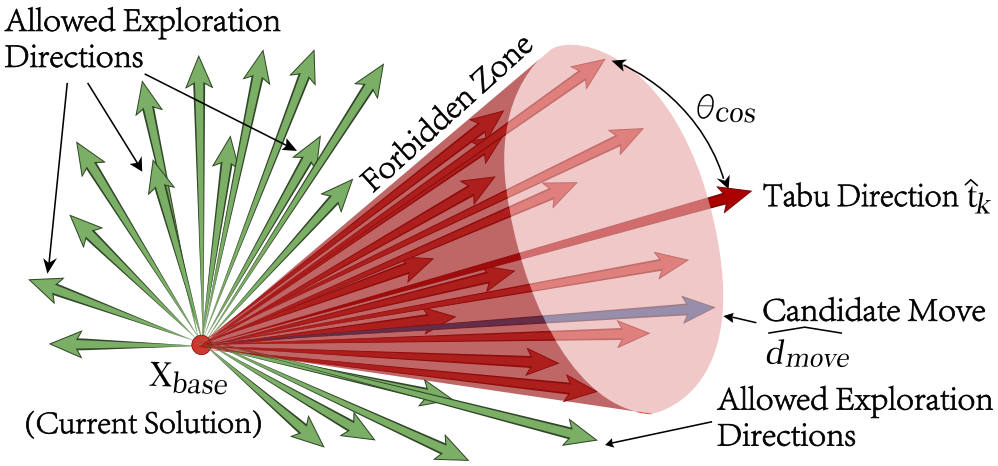


Fig. 5: Schematic illustration of the population-based directional tabu mechanism for exploration.

$$M = \frac{\|X_{\text{candidate}} - X_{\text{base}}\|_2}{\text{NormRange}} \tag{16}$$

$$I = \frac{f(X_{\text{base}}) - f(X_{\text{candidate}})}{|f(X_{\text{base}})| + \epsilon} \tag{17}$$

$$IF = w \cdot M + (1 - w) \cdot I \tag{18}$$

To achieve an adaptive balance between memory retention and search flexibility, both the decision to record a search direction and the duration of its tabu prohibition are governed by an Improvement Strength Factor (IF). This factor jointly evaluates the displacement magnitude and the relative fitness gain induced by a move. The normalized step length M and the relative fitness improvement I are defined in Eq. 16 and Eq. 17, respectively, where Norm Range represents the diagonal length of the search space, and ϵ is a small constant for numerical stability. The Improvement Strength Factor is then synthesized as a weighted combination in Eq. 18, where $w \in [0, 1]$ regulates the relative emphasis between exploration step size and fitness gain.

$$T_{\text{tenure}} = L_{\min} + \text{round}((L_{\max} - L_{\min}) \cdot (1 - IF)) \tag{19}$$

Subsequently, DT-DTT determines the Tabu Tenure T_{tenure} adaptively via an inverse regulation strategy driven by IF. The core rationale is to penalize inefficiency: directions associated with weak improvements ($IF \to 0$) are assigned longer tenures to strictly suppress repetition, whereas directions yielding substantial gains ($IF \to 1$) are treated conservatively with shorter or zero tenure to allow continued exploitation. Defined in Eq. 19, this adaptive tabu tenure mechanism ensures that the tabu memory is dominated by truly non-productive directions without imposing excessive restrictions on promising search behaviors.

Through this coordinated design, DT-DTT provides the PDMSSA with a state-dependent local escape mechanism. It discourages redundant exploration during stagnation phases while supporting precise refinement when the search progresses toward promising regions, thereby regulating the exploration-exploitation trade-off in a controlled and adaptive manner. The overall execution procedure of the proposed strategy is summarized in Algorithm 2.

Algorithm 2 Pseudocode of SADTSM

Require: Current population state $\mathbf{X}$, Global best $\mathbf{X}_{food}$, Max TS iterations T_{TS}.
Ensure: Refined global best $\mathbf{X}_{food}$.

Initialize Tabu List $\mathcal{L} \leftarrow \emptyset$ and Momentum vector $\mathbf{V}_{mom} \leftarrow \mathbf{0}$;
Determine adaptive tabu threshold θ_{cos} based on population spatial diversity;
Set current base solution $\mathbf{X}_{base} \leftarrow \mathbf{X}_{food}$;
for $t = 1$ to T_{TS} **do**
    / Step 1: Hybrid Candidate Generation */*
    Generate candidates $\mathbf{X}_{cand1}$ and $\mathbf{X}_{cand2}$ based on Eq. 9 and Eq. 10;
    Evaluate fitness and select the optimal $\mathbf{X}_{cand}$ based on greedy criterion Eq. 14;
    Calculate normalized movement direction $\mathbf{d} \leftarrow \frac{\mathbf{X}_{cand} - \mathbf{X}_{base}}{\|\mathbf{X}_{cand} - \mathbf{X}_{base}\|_2}$;
    / Step 2: Directional Tabu Check */*
    $isTabu \leftarrow$ **True** if $\mathbf{d}$ falls into any forbidden cone in $\mathcal{L}$ (via θ_{cos});
    $isAspiration \leftarrow$ **True** if $f(\mathbf{X}_{cand}) < f(\mathbf{X}_{food})$;
    if (**not** $isTabu$) **or** $isAspiration$ **then**
        if $f(\mathbf{X}_{cand}) < f(\mathbf{X}_{base})$ **then**
            Calculate Improvement Factor (IF) based on Eq. 18;
            / Step 3: Dynamic Update Mechanism */*
            if IF indicates significant improvement ($IF \geq \theta_{skip}$) **then**
                Super-Exemption: Skip adding $\mathbf{d}$ to $\mathcal{L}$;
                Strengthen Momentum $\mathbf{V}_{mom}$ for inertia utilization;
            else
                Determine adaptive tabu tenure L based on IF via Eq. 19;
                Add $\{\mathbf{d}, t + L\}$ to Tabu List $\mathcal{L}$;
                Maintain normal Momentum $\mathbf{V}_{mom}$;
            end if
            Update $\mathbf{X}_{base} \leftarrow \mathbf{X}_{cand}$;
            if $f(\mathbf{X}_{cand}) < f(\mathbf{X}_{food})$ **then**
                Update $\mathbf{X}_{food} \leftarrow \mathbf{X}_{cand}$;
            end if
        end if
    else
        Stagnation Handling: Decay Momentum $\mathbf{V}_{mom}$ to suppress oscillation;
    end if
    Remove expired direction records from $\mathcal{L}$;
end for
return $\mathbf{X}_{food}$;

3.4 Remarks

To address the limitations of the SSA in complex optimization landscapes, such as insufficient exploration capability, premature convergence, and limited convergence precision in later stages, this study proposes three core refinement strategies to boost the performance of SSA, and the refined algorithm named PDMSSA. The proposed framework enhances the search process by coordinating complementary mechanisms that target different structural weaknesses of the original SSA.

Firstly, to alleviate the over-reliance on a single fitness criterion in leader selection, which often results in highly concentrated leader distributions, a fast non-dominated sorting–based leader selection mechanism is introduced. By jointly considering solution fitness and relative distance, this mechanism preserves leader quality while explicitly maintaining spatial diversity, allowing multiple leaders to guide the population along distinct search directions. At second, to overcome the limitations of single-path information propagation in follower updates, a scale-free network–based multi-source information fusion mechanism is incorporated. By mapping population individuals onto a scale-free topology and integrating neighborhood-based guidance with dual-elite cooperation, followers can aggregate informative cues from both local and global sources. This design retains the chain-based inheritance of SSA while improving information dissemination efficiency and responsiveness during evolution. At last, to address the absence of directional memory and the repeated exploration of ineffective paths in later search stages, a state-aware directional tabu search strategy is embedded into the framework. By recording and regulating historically inefficient exploration directions at the vector level, and by jointly considering trajectory characteristics and fitness evolution, the proposed strategy enables adaptive detection of stagnation states and targeted local intervention without restricting specific solution positions.

Through the coordinated integration of these components, PDMSSA establishes an adaptive collaborative search framework that regulates exploration and exploitation in a complementary manner. The corresponding algorithm pseudocode is provided in Algorithm 3, and the overall workflow of the proposed method is illustrated in Fig. 6.

3.5 Computational Complexity Analysis

Let the population size be N, the problem dimensionality be D, the maximum number of iterations be T, and the computational cost of a single objective-function evaluation be C. During the initialization stage, the positions of all individuals are randomly generated and evaluated once. The computational overhead mainly comes from generating D-dimensional position vectors and performing objective-function evaluations, resulting in a time complexity of $O\left(N(D+C)\right)$.

In the main evolutionary process, the algorithm executes a total of T iterations. At each iteration, the fast non-dominated sorting–based leader selection mechanism performs Pareto ranking with two objectives, yielding a worst-case time complexity of $O(N^2)$. The scale-free network–based multi-source information fusion strategy involves periodic network construction with a worst-case cost of $O(N^2)$; since this operation is not executed at every iteration, its amortized contribution remains limited. During follower updates, each individual conducts a bounded neighborhood exploration on

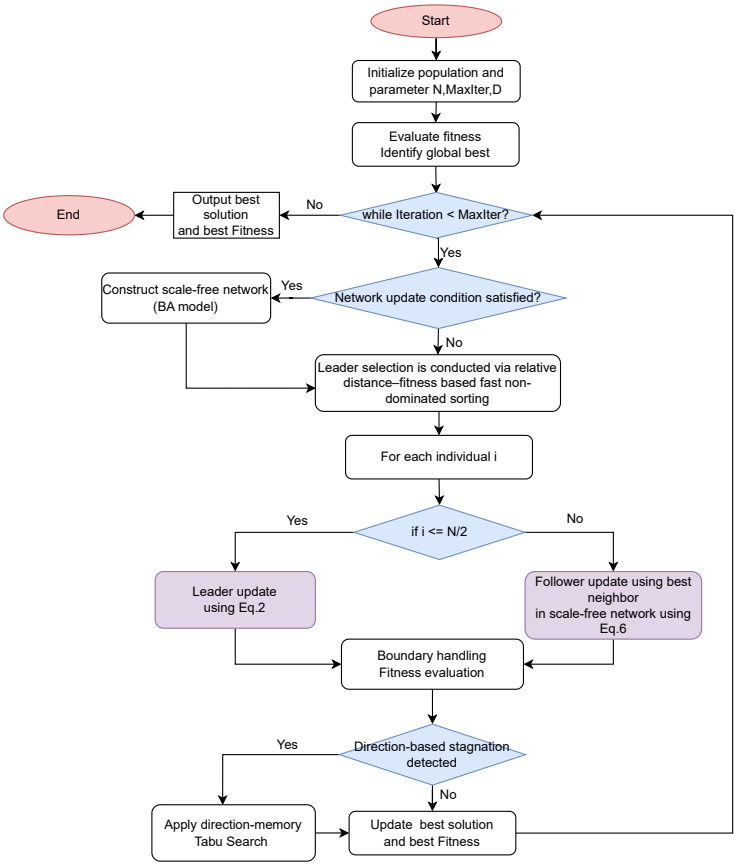


Fig. 6: Overall flowchart of the proposed PDMSSA.

the network topology using breadth-first search, leading to an overall cost of $O(N^2)$ for the population. Population position updates for both leaders and followers require D-dimensional vector operations, incurring a cost of $O(ND)$, followed by objective-function evaluations for all individuals with a cost of $O(NC)$.

The state-aware directional tabu search module is activated only when predefined triggering conditions are satisfied. In a single activation, neighborhood solution generation and evaluation introduce a computational cost proportional to $O(N(D+C))$. Given that the internal tabu search iterations are bounded and that this module is not executed at every generation, its contribution does not alter the overall complexity order. Consequently, the worst-case time complexity of a single iteration is dominated by the non-dominated sorting and network-related operations, resulting in $O(N^2 + ND + NC)$. Over T iterations, the overall time complexity of PDMSSA is therefore $O\left(T(N^2 + ND + NC)\right)$.

Compared with the original salp swarm algorithm (SSA), whose time complexity is $O(T(ND + NC))$, PDMSSA introduces an additional $O(N^2)$ computational term.

Algorithm 3 Pseudocode of the Proposed PDMSSA

Require: Population size N, Dimension D, Max iterations T_{max}, Objective function $f(\cdot)$.
Ensure: Global optimal position $\mathbf{X}_{food}$ and its fitness F_{food}.

Initialize population $\mathbf{X} \in \mathbb{R}^{N \times D}$ and evaluate fitness **Fit**;
Identify global best $\mathbf{X}_{food}$ and local best $\mathbf{X}_{Pbest}$;
Initialize auxiliary memory structures for state monitoring;
Set current iteration $t \leftarrow 1$;
while $t \leq T_{max}$ **do**
    Update time-varying parameter c_1 based on Eq. 3;
    / Phase 1: FNSLSM Strategy */*
    $[\mathbf{idx}_L, \mathbf{idx}_F] \leftarrow \text{ParetoSort}(\mathbf{X}, \mathbf{Fit})$;
    Reorder population $\mathbf{X}$ based on the sorting results;
    / Phase 2: SFNMIFS Strategy */*
    if $t \pmod{T_{update}} == 0$ **then** ▷ Periodic topology update
        $\mathcal{G}_{SFN} \leftarrow \text{BuildScaleFreeNetwork}(N)$;
    end if
    / Position Update Phase */*
    for $i = 1$ to N **do**
        if $i \in \mathbf{idx}_L$ **then** ▷ Leader Phase
            Update $\mathbf{X}_i$ based on Eq. 2;
        else ▷ Follower Phase (Multi-source Fusion)
            $\mathbf{X}_{neigh} \leftarrow \text{FindBestNeighbor}(i, \mathcal{G}_{SFN}, \mathbf{X})$;
            Update $\mathbf{X}_i$ based on Eq. 6;
        end if
        Check boundary constraints $[lb, ub]$;
        Update fitness $Fit_i \leftarrow f(\mathbf{X}_i)$;
    end for
    Update global best $\mathbf{X}_{food}$ and local best $\mathbf{X}_{Pbest}$;
    / Phase 3: SADTSM Strategy */*
    if $TI < \epsilon_{TI}$ **and** $\Delta f < \epsilon_f$ **then** ▷ Stagnation detection
        $[\mathbf{X}_{food}, F_{food}] \leftarrow \text{TabuSearch}(\mathbf{X}_{food}, \mathcal{TL}, \dots)$;
        Reset monitoring states and update dynamic tabu tenure;
    end if
    $t \leftarrow t + 1$;
end while
return $\mathbf{X}_{food}, F_{food}$;

In high-dimensional optimization scenarios where $D \gg N$, the ND term dominates the overall computational cost, and the time complexity of PDMSSA remains in the same order of magnitude as that of the original SSA. When the population size N becomes relatively large, the $O(N^2)$ overhead introduced by the Pareto-based sorting procedure and the network-based neighborhood exploration leads to an increased computational burden. However, in typical applications of population-based metaheuristic algorithms, the population size is usually kept within a moderate range (e.g., 30–100), under which this additional cost remains manageable. Consequently, PDMSSA preserves computational tractability while incorporating an $O(N^2)$ overhead to support enhanced search mechanisms.

4 Experiment and performance evaluation

4.1 Experiment setup

To comprehensively evaluate the performance of the proposed algorithm under varying levels of problem complexity and dimensionality, two widely adopted benchmark test suites, CEC2017 and CEC2022, are employed. Both suites were developed by the IEEE Congress on Evolutionary Computation (CEC) competition committee and follow standardized function construction principles with known global optima. They are commonly used to assess global search capability, convergence accuracy, and scalability with respect to problem dimensionality.

The CEC2017 benchmark suite consists of 30 test functions categorized into four groups according to their landscape characteristics. The unimodal functions include F1 and F3 (F2 is excluded from the final suite). These functions contain a single global optimum and no local optima, and are primarily used to evaluate convergence speed and exploitation ability. The simple multimodal functions (F4–F10) contain numerous local optima, with their number increasing rapidly as dimensionality grows, making them suitable for assessing global exploration capability and the ability to escape local optima. The hybrid functions (F11–F20) are constructed by combining multiple basic functions through weighting, permutation, rotation, and shifting operations. These functions are typically non-separable and exhibit strong variable interactions, posing challenges related to multi-scale search and variable coupling. The composition functions (F21–F30) further increase landscape complexity by embedding different basic functions into distinct regions of the search space and incorporating additional rotation and shift transformations. This group is widely used to evaluate algorithmic robustness and stability in highly deceptive and nonlinear environments. Overall, the CEC2017 suite provides a progressive increase in problem difficulty, enabling a systematic evaluation of exploration–exploitation balance, convergence behavior, and solution stability.

The CEC2022 benchmark suite provides a standardized set of 12 test functions for single-objective continuous optimization with bounded search spaces. These functions are systematically grouped into four categories based on their construction characteristics, offering complementary evaluation scenarios to further examine algorithm performance across diverse landscape structures. Specifically, F1 is a unimodal function with a unique global optimum and is primarily used to evaluate local exploitation capability and convergence speed. Functions F2–F5 are categorized as basic functions. Despite this designation, they generally exhibit multimodal characteristics and are therefore suitable for assessing global exploration ability and the capacity to escape local optima. Functions F6–F8 are hybrid functions constructed by shuffling and combining multiple basic subfunctions. The resulting components are typically non-separable, making this group appropriate for examining adaptability to variable coupling and heterogeneous landscape structures. Functions F9–F12 are composition functions that integrate multiple function characteristics across different regions of the search space and incorporate rotation and shift transformations, yielding highly complex and deceptive fitness landscapes. This group is mainly used to evaluate global optimization capability and robustness.

To facilitate comparative performance evaluation, a total of 29 and 12 benchmark functions are selected from the CEC2017[56] and CEC2022[57] test suites, respectively. The CEC2017 suite covers a broad range of continuous optimization problems, from unimodal to highly multimodal landscapes with varying degrees of variable interaction. In the experiments, problem dimensionalities of 50 and 100 are considered to investigate stability and scalability in high-dimensional settings. In contrast, the 12 functions in the CEC2022 suite are evaluated under 10-dimensional and 20-dimensional configurations, providing complementary assessment of search efficiency and convergence behavior in medium-dimensional problems.

To assess the effectiveness of the proposed PDMSSA within the same algorithmic family, four representative variants of the salp swarm algorithm (SSA), namely MSNSSA[32], HSSAPSO[36], TLSSA[43], and DMSSA[45], are selected as comparison methods. In addition, five recently proposed metaheuristic optimization algorithms—SOFA[18], COA[19], DBO[20], GGO[21], and CPO[23]—are included to further examine the competitiveness of PDMSSA in a broader algorithmic context. The key parameter settings of all comparison algorithms are summarized in Table 2.

All experiments were conducted on the OpenSCOW HPC platform using a single compute node equipped with two Intel(R) Xeon(R) Gold 6348 processors (2.60 GHz, 56 CPU cores in total), with MATLAB R2022a. To ensure experimental fairness, the population size of all algorithms was set to 50, and the maximum number of iterations was fixed at 1000. Each algorithm was independently executed 30 times under identical parameter configurations to reduce the influence of stochastic effects. For the considered minimization problems, algorithm performance is primarily evaluated using the mean of the best objective values obtained over the 30 independent runs, where smaller mean values indicate better overall optimization performance. When mean values are comparable, the standard deviation is further examined, with lower standard deviations reflecting greater stability and robustness across repeated runs.

Table 2: Parameter settings of the comparison algorithms

Algorithm	Year	Parameter Settings
SSA	2017	$c_1 = 2e^{-\left(\frac{4t}{T_{\max}}\right)^2}, c_2, c_3 \in [0,1]$
MSNSSA	2022	$m = 2.5, R \in \{1,2\}$
HSSAPSO	2022	$c_1 = 2e^{-\left(\frac{4t}{T_{\max}}\right)^2}, c_1^p = c_2^p = 1.5$
DMSSA	2023	$CR \sim \mathcal{N}(0.5, 0.1), F \sim \mathcal{C}(0.5, 0.1), c = 0.1$
DBO	2022	$k = 0.1, b = 0.3, \alpha \in \{-1, +1\}$
COA	2023	$I \in 1, 2, r \in [0,1]$
GGO	2024	$\omega_{1\sim4} \in [0,2], r_{1\sim5} \in [0,1]$
CPO	2025	$Q = 100, u = 2 + r_1, H = \frac{r_2}{2}, r_1, r_2 \in [0,1]$
SFOA	2025	$r \in [0,1], s = 20r_1 + 20r_2, r_1, r_2 \sim \mathcal{N}(0,1)$
TLSSA	2023	$c_1 = 2e^{-\left(\frac{4t}{T_{\max}}\right)^2}, TF \in 1, 2, \beta = \frac{3}{2}$
PDMSSA	/	$c_1 = 2e^{-\left(\frac{4t}{T_{\max}}\right)^2}, c_2, c_3 \in [0,1]$

4.2 Convergence Accuracy Comparison

In this subsection, the convergence accuracy of the proposed PDMSSA is comparatively analyzed against several benchmark algorithms on the CEC2017 and CEC2022 test suites. Table 3 and Table 4 report the experimental results on the CEC2017 benchmark under 50-dimensional and 100-dimensional settings, respectively, while Table 5 and Table 6 present the results on the CEC2022 benchmark for 10-dimensional and 20-dimensional cases.

To reduce the influence of occasional abnormal runs on the overall conclusions, the mean of the best objective values obtained over 30 independent runs is adopted as the performance metric for evaluating convergence accuracy, thereby ensuring the robustness and representativeness of the results. For clarity in comparison, the best-performing results (i.e., the smallest mean values) in each table are highlighted in bold font.

According to the statistical results reported in Table 3, under the 50-dimensional setting of the CEC2017 benchmark, PDMSSA ranks first with an overall mean rank of 1.24, indicating superior overall optimization performance compared with the competing algorithms under this test condition.

For the unimodal functions F1 and F3, PDMSSA achieves the best performance on F3 and ranks second on F1. The corresponding mean values and standard deviations are comparable to those of the leading algorithms, indicating reliable exploitation capability and stable solution accuracy in high-dimensional unimodal problems. For the simple multimodal functions F4–F10, PDMSSA ranks first on all seven test functions. Notably, on more challenging cases such as F6 and F9, which contain numerous irregular local optima, PDMSSA still attains near-optimal mean results, demonstrating strong global search effectiveness and resistance to premature convergence. In the hybrid function group (F11–F20), PDMSSA achieves first place on six functions and ranks within the top two on F13, F14, and F20. Given the strong variable coupling and non-separable structures of these problems, the results indicate good adaptability to complex search landscapes. Although PDMSSA exhibits a relatively lower ranking on F18, its mean performance remains competitive, without notable degradation compared with the best-performing algorithms. For the composition functions F21–F30, which integrate rotation, shift, and multimodality, PDMSSA achieves the best performance on eight functions (F21–F27 and F29) and ranks second on F28. Even on particularly difficult functions such as F23 and F24, its results remain among the top-performing methods. Overall, the results in Table X show that PDMSSA maintains a consistent balance between solution quality and robustness across diverse high-dimensional optimization problems.

To further examine the scalability and robustness of PDMSSA in large-scale search spaces, the problem dimensionality is increased to 100. As reported in Table 4, PDMSSA maintains a leading position under this high-dimensional setting, achieving an overall mean rank of 1.14. This result indicates that the algorithm preserves stable overall performance as dimensionality increases, without evident degradation. For the unimodal functions F1 and F3, PDMSSA achieves the best ranking on both functions, demonstrating consistent solution accuracy in enlarged search spaces. This

Table 3: Comparison results on the CEC2017 benchmark functions (50D)

Function	Index	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO	PDMSSA
F1	Mean	8.17E+03	**2.51E+03**	6.91E+03	3.48E+05	2.54E+04	1.90E+11	7.08E+10	3.89E+08	1.08E+11	2.32E+06	4.26E+03
	Std	1.00E+04	2.66E+03	6.39E+03	4.25E+05	2.12E+04	4.19E+10	1.20E+10	2.11E+08	9.81E+09	1.45E+06	2.52E+03
	Rank	4	1	3	6	5	11	9	8	10	7	2
F3	Mean	1.31E+05	6.15E+04	2.91E+05	1.88E+05	1.16E+05	6.75E+06	2.35E+05	2.26E+05	1.89E+05	3.10E+05	**1.36E+04**
	Std	4.20E+04	1.33E+04	7.84E+04	3.74E+04	2.36E+04	2.05E+07	6.27E+04	5.67E+04	1.99E+04	8.61E+04	4.96E+03
	Rank	4	2	9	5	3	11	8	7	6	10	1
F4	Mean	6.12E+02	6.09E+02	6.34E+02	**5.75E+02**	6.46E+02	7.63E+04	1.82E+04	8.05E+02	3.65E+04	6.56E+02	5.82E+02
	Std	5.07E+01	5.97E+01	4.73E+01	4.25E+01	4.30E+01	2.23E+04	3.09E+03	1.99E+02	4.44E+03	4.14E+01	4.14E+01
	Rank	4	3	5	1	6	11	9	8	10	7	2
F5	Mean	8.29E+02	6.71E+02	7.94E+02	8.30E+02	8.58E+02	1.56E+03	1.05E+03	9.24E+02	1.19E+03	8.76E+02	**5.60E+02**
	Std	6.88E+01	3.42E+01	5.00E+01	9.05E+01	5.36E+01	1.08E+02	4.43E+01	9.40E+01	2.51E+01	1.64E+01	1.85E+01
	Rank	4	2	3	5	6	11	9	8	10	7	1
F6	Mean	6.60E+02	6.22E+02	6.52E+02	6.23E+02	6.66E+02	7.39E+02	6.75E+02	6.59E+02	6.99E+02	6.72E+02	**6.00E+02**
	Std	9.05E+00	3.84E+00	8.06E+00	1.10E+01	7.06E+00	1.48E+01	8.14E+00	8.53E+00	5.16E+00	5.72E+00	5.52E-02
	Rank	6	2	4	3	7	11	9	5	10	8	1
F7	Mean	1.20E+03	9.38E+02	1.21E+03	1.18E+03	1.35E+03	4.77E+03	1.77E+03	1.24E+03	2.02E+03	1.61E+03	**8.24E+02**
	Std	1.09E+02	4.14E+01	1.51E+02	6.43E+01	2.03E+02	8.17E+02	1.22E+02	1.36E+02	5.35E+01	1.61E+02	2.37E+01
	Rank	4	2	5	3	7	11	9	6	10	8	1
F8	Mean	1.13E+03	9.61E+02	1.11E+03	1.10E+03	1.20E+03	1.88E+03	1.36E+03	1.23E+03	1.48E+03	1.22E+03	**8.59E+02**
	Std	7.38E+01	2.78E+01	5.95E+01	8.87E+01	5.73E+01	9.90E+01	3.89E+01	9.13E+01	2.88E+01	2.90E+01	1.57E+01
	Rank	5	2	4	3	6	11	9	8	10	7	1
F9	Mean	1.38E+04	2.79E+03	1.20E+04	1.14E+04	1.38E+04	8.17E+04	2.14E+04	1.97E+04	3.62E+04	3.06E+04	**9.02E+02**
	Std	2.88E+03	7.12E+02	2.99E+03	4.60E+03	1.79E+03	1.46E+04	5.01E+03	6.28E+03	2.69E+03	1.18E+04	1.43E+00
	Rank	6	2	4	3	5	11	8	7	10	9	1
F10	Mean	7.86E+03	6.93E+03	8.72E+03	1.32E+04	8.09E+03	1.68E+04	1.26E+04	9.48E+03	1.51E+04	1.06E+04	**5.81E+03**
	Std	1.15E+03	1.15E+03	1.36E+03	5.64E+02	1.26E+03	5.50E+02	7.86E+02	1.71E+03	3.85E+02	3.62E+03	1.03E+03
	Rank	3	2	5	9	4	11	8	6	10	7	1
F11	Mean	1.55E+03	1.49E+03	1.96E+03	1.41E+03	1.92E+03	8.61E+04	2.33E+04	2.08E+03	2.54E+04	1.67E+03	**1.31E+03**
	Std	8.62E+01	6.68E+01	3.09E+02	7.31E+01	2.02E+02	3.45E+04	7.77E+03	3.28E+02	2.02E+03	1.76E+02	6.52E+01
	Rank	4	3	7	2	6	11	9	8	10	5	1
F12	Mean	9.17E+07	6.33E+07	9.93E+07	3.94E+06	1.57E+08	9.76E+10	3.69E+10	4.10E+08	8.06E+10	5.06E+07	**3.75E+06**
	Std	4.62E+07	3.54E+07	5.40E+07	3.50E+06	1.25E+08	2.56E+10	8.29E+09	4.29E+08	1.54E+10	3.10E+07	1.79E+06
	Rank	5	4	6	2	7	11	9	8	10	3	1
F13	Mean	1.48E+05	8.04E+04	1.95E+05	**1.61E+04**	1.34E+05	4.87E+10	1.91E+10	2.30E+07	4.54E+10	9.76E+04	2.97E+04
	Std	9.51E+04	4.11E+04	1.14E+05	1.08E+04	7.75E+04	1.90E+10	7.47E+09	3.02E+07	1.17E+10	5.24E+04	1.10E+04
	Rank	6	3	7	1	5	11	9	8	10	4	2
F14	Mean	1.82E+05	1.99E+05	4.44E+05	**6.10E+03**	6.21E+05	1.23E+08	3.71E+07	1.51E+06	6.50E+07	7.64E+05	3.58E+04
	Std	1.38E+05	1.64E+05	3.14E+05	1.32E+04	3.35E+05	8.12E+07	2.54E+07	2.48E+06	4.53E+07	5.35E+05	2.31E+04
	Rank	3	4	5	1	6	11	9	8	10	7	2
F15	Mean	6.75E+04	3.44E+04	1.09E+05	**8.61E+03**	4.98E+04	1.60E+10	2.29E+09	1.41E+06	8.31E+09	3.00E+04	1.02E+04
	Std	3.46E+04	1.77E+04	6.21E+04	6.93E+03	4.14E+04	8.31E+09	1.79E+09	5.34E+06	2.58E+09	2.16E+04	4.88E+03
	Rank	6	4	7	1	5	11	9	8	10	3	2
F16	Mean	3.76E+03	3.30E+03	3.92E+03	3.89E+03	4.24E+03	1.12E+04	6.24E+03	4.56E+03	9.49E+03	4.76E+03	**2.41E+03**
	Std	4.55E+02	3.87E+02	5.42E+02	7.65E+02	5.64E+02	2.40E+03	7.22E+02	5.38E+02	1.35E+03	1.16E+03	3.16E+02
	Rank	3	2	5	4	6	11	9	7	10	8	1
F17	Mean	3.44E+03	3.08E+03	3.37E+03	3.28E+03	3.56E+03	2.88E+05	4.31E+03	4.21E+03	7.82E+03	3.64E+03	**2.52E+03**
	Std	3.58E+02	2.76E+02	3.74E+02	4.48E+02	4.09E+02	4.80E+05	5.67E+02	4.97E+02	2.60E+03	4.39E+02	3.04E+02
	Rank	5	2	4	3	6	11	9	8	10	7	1
F18	Mean	3.14E+06	1.21E+06	4.44E+06	5.45E+05	3.66E+06	4.00E+08	7.18E+07	8.15E+06	1.61E+08	2.79E+06	**3.76E+05**
	Std	2.38E+06	7.13E+05	3.33E+06	1.48E+06	3.09E+06	2.71E+08	4.80E+07	8.33E+06	8.64E+07	2.14E+06	2.21E+05
	Rank	5	3	7	2	6	11	9	8	10	4	1
F19	Mean	4.12E+06	4.06E+05	5.78E+06	**1.25E+04**	5.20E+06	7.81E+09	8.79E+08	5.60E+06	2.90E+09	3.67E+05	1.33E+04
	Std	2.15E+06	5.24E+05	3.82E+06	1.29E+04	4.65E+06	3.14E+09	6.98E+08	4.70E+06	1.45E+09	5.24E+05	7.51E+03
	Rank	5	4	8	1	6	11	9	7	10	3	2
F20	Mean	3.35E+03	2.99E+03	3.33E+03	3.57E+03	3.24E+03	5.31E+03	3.82E+03	3.53E+03	4.14E+03	3.76E+03	**2.64E+03**
	Std	3.60E+02	3.18E+02	3.94E+02	2.80E+02	3.29E+02	3.84E+02	2.81E+02	3.42E+02	2.16E+02	1.64E+02	2.76E+02
	Rank	5	2	4	7	3	11	9	6	10	8	1
F21	Mean	2.59E+03	2.44E+03	2.59E+03	2.57E+03	2.73E+03	3.43E+03	2.94E+03	2.79E+03	3.22E+03	2.85E+03	**2.36E+03**
	Std	8.29E+01	2.90E+01	4.71E+01	8.92E+01	1.08E+02	1.20E+02	6.16E+01	7.58E+01	7.96E+01	9.41E+01	1.46E+01
	Rank	5	2	4	3	6	11	9	7	10	8	1
F22	Mean	9.40E+03	**6.04E+03**	9.82E+03	1.46E+04	1.07E+04	1.85E+04	1.45E+04	1.08E+04	1.68E+04	1.30E+04	6.63E+03
	Std	1.06E+03	3.38E+03	1.12E+03	5.75E+02	1.38E+03	4.08E+02	8.90E+02	1.23E+03	7.83E+02	3.32E+03	2.14E+03
	Rank	3	1	4	9	5	11	8	6	10	7	2
F23	Mean	3.03E+03	2.89E+03	3.05E+03	3.04E+03	3.17E+03	4.40E+03	4.35E+03	3.37E+03	4.47E+03	3.88E+03	**2.78E+03**
	Std	8.01E+01	3.39E+01	6.87E+01	9.74E+01	1.22E+02	2.92E+02	2.40E+02	8.31E+01	1.62E+02	2.11E+02	2.59E+01
	Rank	3	2	5	4	6	10	9	7	11	8	1
F24	Mean	3.15E+03	3.02E+03	3.17E+03	3.20E+03	3.31E+03	4.34E+03	4.84E+03	3.52E+03	4.77E+03	4.36E+03	**2.96E+03**
	Std	5.18E+01	2.98E+01	6.19E+01	8.93E+01	9.88E+01	2.71E+02	2.47E+02	1.43E+02	1.73E+02	3.56E+02	2.14E+01
	Rank	3	2	4	5	6	8	11	7	10	9	1
F25	Mean	3.08E+03	3.10E+03	3.10E+03	3.09E+03	3.13E+03	4.17E+04	1.04E+04	3.41E+03	1.50E+04	3.15E+03	**3.05E+03**
	Std	3.12E+01	2.41E+01	2.97E+01	2.76E+01	3.89E+01	1.20E+04	9.55E+02	1.05E+03	1.26E+03	2.34E+01	2.22E+01
	Rank	2	5	4	3	6	11	9	8	10	7	1
F26	Mean	4.57E+03	5.04E+03	6.49E+03	6.65E+03	4.61E+03	2.38E+04	1.47E+04	1.03E+04	1.72E+04	1.02E+04	**4.19E+03**
	Std	1.98E+03	2.60E+03	1.33E+03	1.12E+03	2.33E+03	3.81E+03	1.11E+03	1.63E+03	6.30E+02	2.44E+03	1.56E+02
	Rank	2	4	5	6	3	11	9	8	10	7	1
F27	Mean	3.60E+03	3.58E+03	3.59E+03	3.42E+03	3.81E+03	6.08E+03	6.55E+03	3.81E+03	6.80E+03	5.08E+03	**3.29E+03**
	Std	1.17E+02	1.29E+02	1.14E+02	1.02E+02	1.64E+02	7.33E+02	4.92E+02	1.78E+02	6.09E+02	6.27E+02	3.58E+01
	Rank	5	3	4	2	7	9	10	6	11	8	1
F28	Mean	3.34E+03	3.35E+03	3.39E+03	3.36E+03	3.46E+03	1.94E+04	9.16E+03	5.27E+03	1.33E+04	3.49E+03	**3.30E+03**
	Std	3.57E+01	3.53E+01	4.27E+01	3.48E+01	6.46E+01	3.53E+03	8.45E+02	2.14E+03	1.36E+03	5.38E+01	2.41E+01
	Rank	2	3	5	4	6	11	9	8	10	7	1
F29	Mean	5.50E+03	5.04E+03	5.52E+03	4.68E+03	6.18E+03	3.21E+05	1.81E+04	5.77E+03	8.92E+04	6.37E+03	**3.81E+03**
	Std	4.77E+02	4.82E+02	5.31E+02	5.08E+02	7.87E+02	5.54E+05	1.13E+04	6.14E+02	7.06E+04	1.15E+03	1.95E+02
	Rank	4	3	5	2	7	11	9	6	10	8	1
F30	Mean	1.00E+08	9.12E+07	1.07E+08	3.40E+06	1.30E+08	1.27E+10	2.30E+09	2.92E+07	7.73E+09	4.85E+07	**2.19E+06**
	Std	3.03E+07	1.73E+07	3.78E+07	1.25E+06	4.78E+07	6.35E+09	1.54E+09	2.78E+07	2.69E+09	1.52E+07	7.59E+05
	Rank	6	5	7	2	8	11	9	3	10	4	1
Rank		0	2	0	5	0	0	0	0	0	0	22
Ave. Rank		4.21	2.72	5.14	3.52	5.69	10.79	8.97	7.07	9.93	6.72	1.24
Total Rank		4	2	5	3	6	11	9	8	10	7	1

suggests that effective exploitation behavior can be maintained even under high-dimensional conditions. For the simple multimodal functions F4–F10, PDMSSA again ranks first on all seven test functions. Despite the substantial increase in the number of local optima with dimensionality, the algorithm consistently attains competitive mean results, indicating robust search behavior in high-dimensional multimodal landscapes. In the hybrid function group (F11–F20), PDMSSA achieves the best performance on seven functions and remains among the top-performing algorithms on the remaining

cases, with limited performance variation. Given the presence of strong variable coupling, rotational transformations, and non-separable structures, these results reflect good adaptability to complex high-dimensional search landscapes. For the composition functions F21–F30, which combine multiple challenging characteristics, PDMSSA attains the best performance on nine functions and ranks second on F30, demonstrating stable overall behavior under highly complex conditions. Overall, the statistical results on the 100-dimensional CEC2017 benchmark indicate that PDMSSA maintains a favorable balance between solution quality and robustness as problem dimensionality increases, exhibiting good scalability in high-dimensional complex optimization problems.

Table 5 reports the experimental results on the CEC2022 benchmark under the 10-dimensional setting. Overall, PDMSSA exhibits stable and competitive performance in this low-dimensional test environment. For the unimodal function F1, PDMSSA ranks second, with mean performance close to that of the best-performing algorithm and a comparable standard deviation, indicating reliable convergence accuracy. For the basic functions F2–F5, PDMSSA achieves the best results on F3, F4, and F5, while its performance on the remaining functions remains among the top-ranked algorithms. These results suggest that PDMSSA maintains balanced search behavior in low-dimensional landscapes with moderate multimodality. In the hybrid function group (F6–F8), PDMSSA attains the best performance on F8 and shows stable results on the other functions, reflecting reasonable adaptability to variable coupling and heterogeneous search structures. For the composition functions F9–F12, PDMSSA achieves the best performance on part of the test functions, while its rankings on others are relatively lower, which is consistent with the increased complexity and higher search difficulty of this function category. Overall, under the 10-dimensional CEC2022 test setting, PDMSSA achieves the lowest final average rank of 1.75 among all compared algorithms, indicating favorable and consistent overall performance across different function types.

Under the 20-dimensional CEC2022 benchmark, the results in Table 6 show that PDMSSA maintains stable overall performance, achieving the lowest final average rank of 1.42 among all compared algorithms. For the unimodal function F1, PDMSSA ranks first. In the basic function group (F2–F5), it achieves the best results on F3, F4, and F5, while remaining competitive on F2. For the hybrid functions F6–F8, PDMSSA attains competitive rankings across the test functions. In the composition function group (F9–F12), it achieves the best performance on F10–F12 and ranks second on F9.

Overall, the experimental results on the CEC2017 and CEC2022 benchmark suites indicate that the proposed algorithm maintains stable and competitive performance across different problem dimensionalities. Its consistent behavior over diverse and complex search landscapes suggests reasonable generality in handling high-dimensional numerical optimization problems. Across different categories of test functions, the algorithm exhibits a balanced search behavior, maintaining an appropriate trade-off between exploration and exploitation. As a result, convergence accuracy is preserved while sensitivity to local optima is limited, indicating good stability and adaptability in large-scale nonlinear optimization tasks.

Table 4: Comparison results on the CEC2017 benchmark functions (100D)

Function	Type	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO	PDMSSA
F1	Mean	1.43E+06	7.64E+05	3.30E+08	4.31E+08	1.06E+09	5.19E+11	2.02E+11	6.38E+10	2.64E+11	4.55E+08	**5.08E+03**
	Std	2.67E+06	6.82E+05	1.52E+08	2.38E+08	3.90E+08	7.42E+10	1.94E+10	6.35E+10	1.07E+10	1.77E+08	**6.13E+03**
	rank	3	2	4	5	7	11	9	8	10	6	1
F3	Mean	4.44E+05	2.63E+05	6.28E+05	5.42E+05	3.22E+05	8.10E+07	5.40E+05	5.53E+05	3.50E+05	4.31E+05	**2.48E+05**
	Std	1.26E+05	2.63E+04	3.22E+05	6.87E+04	3.57E+04	2.20E+08	6.69E+04	1.83E+05	**1.21E+04**	1.51E+05	3.87E+04
	rank	6	2	10	8	3	11	7	9	4	5	1
F4	Mean	9.07E+02	9.34E+02	1.07E+03	1.06E+03	1.20E+03	2.28E+05	5.79E+04	3.15E+03	1.06E+05	1.25E+03	**6.91E+02**
	Std	7.20E+01	7.93E+01	1.08E+02	1.06E+02	1.34E+02	5.52E+04	1.04E+04	3.34E+03	1.24E+04	1.51E+02	**4.07E+01**
	rank	2	3	5	4	6	11	9	8	10	7	1
F5	Mean	1.34E+03	1.02E+03	1.28E+03	1.40E+03	1.46E+03	2.84E+03	1.89E+03	1.60E+03	2.11E+03	1.39E+03	**6.82E+02**
	Std	1.02E+02	7.90E+01	8.64E+01	1.52E+02	5.53E+01	2.43E+02	1.00E+02	2.53E+02	**2.98E+01**	3.40E+01	3.05E+01
	rank	4	2	3	6	7	11	9	8	10	5	1
F6	Mean	6.70E+02	6.42E+02	6.66E+02	6.54E+02	6.74E+02	7.57E+02	6.93E+02	6.75E+02	7.10E+02	6.78E+02	**6.01E+02**
	Std	5.05E+00	5.32E+00	4.95E+00	7.47E+00	4.60E+00	1.38E+01	6.68E+00	1.17E+01	3.27E+00	5.52E+00	**3.85E-01**
	rank	5	2	4	3	6	11	9	7	10	8	1
F7	Mean	2.18E+03	1.61E+03	2.30E+03	2.19E+03	2.64E+03	1.11E+04	3.73E+03	2.50E+03	3.97E+03	2.98E+03	**1.01E+03**
	Std	2.89E+02	1.69E+02	2.00E+02	1.46E+02	3.67E+02	1.53E+03	2.69E+02	2.37E+02	8.36E+01	2.52E+02	**4.87E+01**
	rank	3	2	5	4	7	11	9	6	10	8	1
F8	Mean	1.69E+03	1.28E+03	1.62E+03	1.74E+03	1.84E+03	3.32E+03	2.30E+03	2.09E+03	2.56E+03	1.91E+03	**9.86E+02**
	Std	1.33E+02	6.22E+01	9.55E+01	1.78E+02	1.06E+02	2.26E+02	7.19E+01	1.60E+02	5.57E+01	4.86E+01	**4.83E+01**
	rank	4	2	3	5	6	11	9	8	10	7	1
F9	Mean	2.85E+04	1.31E+04	2.86E+04	5.29E+04	3.34E+04	2.03E+05	5.84E+04	5.20E+04	7.82E+04	6.42E+04	**1.05E+03**
	Std	3.15E+03	2.22E+03	2.87E+03	1.19E+04	3.10E+03	3.03E+04	9.00E+03	1.80E+04	4.38E+03	2.24E+04	**6.76E+01**
	rank	3	2	4	7	5	11	8	6	10	9	1
F10	Mean	1.62E+04	1.45E+04	1.65E+04	2.95E+04	1.77E+04	3.47E+04	2.80E+04	2.02E+04	3.24E+04	1.77E+04	**1.29E+04**
	Std	1.83E+03	1.43E+03	2.46E+03	**6.85E+02**	1.75E+03	1.01E+03	9.49E+02	3.30E+03	7.11E+02	1.43E+03	1.69E+03
	rank	3	2	4	9	6	11	8	7	10	5	1
F11	Mean	2.57E+04	1.38E+04	1.02E+05	5.47E+04	5.37E+04	1.10E+06	2.18E+05	1.39E+05	2.34E+05	1.48E+05	**3.22E+03**
	Std	8.48E+03	4.34E+03	3.19E+04	2.31E+04	1.72E+04	5.88E+05	8.41E+04	5.38E+04	3.84E+04	4.36E+04	**2.89E+02**
	rank	3	2	6	5	4	11	9	7	10	8	1
F12	Mean	6.87E+08	5.20E+08	7.17E+08	8.08E+07	1.05E+09	2.73E+11	1.23E+11	2.56E+09	1.99E+11	5.64E+08	**4.08E+07**
	Std	2.76E+08	1.83E+08	3.50E+08	3.62E+07	4.46E+08	5.35E+10	1.71E+10	9.17E+08	1.92E+10	2.54E+08	**1.83E+07**
	rank	5	3	6	2	7	11	9	8	10	4	1
F13	Mean	9.14E+04	5.89E+04	3.30E+05	3.90E+04	1.13E+05	7.50E+10	2.52E+10	1.49E+08	4.47E+10	7.92E+04	**3.66E+04**
	Std	3.41E+04	2.62E+04	8.71E+05	2.43E+04	2.18E+05	1.89E+10	5.76E+09	1.68E+08	5.74E+09	3.32E+04	**9.52E+03**
	rank	5	3	7	2	6	11	9	8	10	4	1
F14	Mean	3.99E+06	1.30E+06	5.61E+06	**4.55E+05**	4.74E+06	4.39E+08	5.62E+07	1.02E+07	8.37E+07	3.52E+06	5.90E+05
	Std	2.34E+06	4.30E+05	2.79E+06	8.90E+05	2.51E+06	3.07E+08	3.32E+07	8.21E+06	3.45E+07	1.83E+06	**3.30E+05**
	rank	5	3	7	1	6	11	9	8	10	4	2
F15	Mean	7.73E+04	4.16E+04	1.27E+05	**1.74E+04**	6.74E+04	3.65E+10	1.06E+10	1.55E+07	2.37E+10	3.44E+05	2.61E+04
	Std	2.90E+04	1.72E+04	1.72E+05	1.83E+04	2.35E+04	9.52E+09	3.46E+09	3.40E+07	4.47E+09	1.66E+06	**8.68E+03**
	rank	5	3	6	1	4	11	9	8	10	7	2
F16	Mean	7.06E+03	5.96E+03	7.10E+03	7.90E+03	8.04E+03	3.18E+04	1.55E+04	8.62E+03	2.40E+04	7.45E+03	**4.48E+03**
	Std	7.09E+02	6.28E+02	8.17E+02	1.85E+03	9.54E+02	7.40E+03	1.94E+03	8.88E+02	2.34E+03	9.22E+02	**6.24E+02**
	rank	3	2	4	6	7	11	9	8	10	5	1
F17	Mean	5.81E+03	5.28E+03	5.70E+03	5.74E+03	5.84E+03	4.15E+07	6.73E+05	8.12E+03	7.93E+06	5.83E+03	**3.94E+03**
	Std	5.73E+02	4.78E+02	5.72E+02	9.02E+02	6.46E+02	5.67E+07	7.91E+05	1.08E+03	5.93E+06	4.92E+02	**4.14E+02**
	rank	5	2	3	4	7	11	9	8	10	6	1
F18	Mean	4.91E+06	1.61E+06	7.88E+06	1.50E+06	4.94E+06	8.28E+08	6.96E+07	1.36E+07	2.26E+08	3.31E+06	**1.14E+06**
	Std	2.94E+06	**5.29E+05**	4.73E+06	1.94E+06	2.26E+06	5.35E+08	3.52E+07	9.86E+06	1.01E+08	2.50E+06	6.11E+05
	rank	5	3	7	2	6	11	9	8	10	4	1
F19	Mean	2.06E+07	9.12E+06	2.06E+07	**1.10E+04**	1.90E+07	3.13E+10	1.00E+10	3.48E+07	2.31E+10	2.19E+06	5.35E+04
	Std	1.06E+07	5.42E+06	1.09E+07	**9.63E+03**	1.33E+07	9.91E+09	2.85E+09	4.84E+07	3.70E+09	1.73E+06	3.07E+04
	rank	7	4	6	1	5	11	9	8	10	3	2
F20	Mean	5.36E+03	4.86E+03	5.53E+03	7.09E+03	5.61E+03	9.34E+03	6.70E+03	6.40E+03	7.74E+03	6.29E+03	**4.26E+03**
	Std	5.25E+02	5.13E+02	5.66E+02	**2.50E+02**	4.53E+02	4.68E+02	5.80E+02	5.56E+02	3.57E+02	1.28E+03	4.90E+02
	rank	3	2	4	9	5	11	8	7	10	6	1
F21	Mean	3.22E+03	2.77E+03	3.15E+03	3.19E+03	3.43E+03	5.19E+03	4.43E+03	3.80E+03	4.95E+03	3.97E+03	**2.49E+03**
	Std	1.53E+02	6.05E+01	9.60E+01	1.70E+02	2.54E+02	2.82E+02	1.71E+02	1.62E+02	1.87E+02	2.37E+02	**2.78E+01**
	rank	5	2	3	4	6	11	9	7	10	8	1
F22	Mean	1.86E+04	1.82E+04	2.01E+04	3.23E+04	2.12E+04	3.71E+04	3.09E+04	2.23E+04	3.47E+04	2.18E+04	**1.52E+04**
	Std	3.38E+03	4.56E+03	1.47E+03	**4.83E+02**	1.99E+03	7.23E+02	1.34E+03	3.67E+03	6.18E+02	1.09E+03	1.30E+03
	rank	3	2	4	9	5	11	8	7	10	6	1
F23	Mean	3.61E+03	3.33E+03	3.70E+03	3.67E+03	3.90E+03	6.05E+03	7.05E+03	4.47E+03	6.55E+03	5.60E+03	**3.00E+03**
	Std	1.20E+02	7.50E+01	1.65E+02	1.70E+02	2.09E+02	4.30E+02	3.96E+02	1.68E+02	3.30E+02	4.63E+02	**4.58E+01**
	rank	3	2	5	4	6	9	11	7	10	8	1
F24	Mean	4.12E+03	3.98E+03	4.22E+03	4.28E+03	4.60E+03	8.50E+03	1.12E+04	5.37E+03	1.02E+04	8.64E+03	**3.40E+03**
	Std	1.99E+02	9.46E+01	1.99E+02	1.58E+02	3.30E+02	9.70E+02	7.13E+02	3.78E+02	7.67E+02	1.14E+03	**2.91E+01**
	rank	3	2	4	5	6	8	11	7	10	9	1
F25	Mean	3.66E+03	3.62E+03	3.83E+03	3.76E+03	3.95E+03	9.57E+04	2.19E+04	7.26E+03	2.84E+04	3.78E+03	**3.34E+03**
	Std	9.77E+01	7.36E+01	9.02E+01	9.89E+01	1.21E+02	2.31E+04	2.88E+03	4.87E+03	1.94E+03	1.04E+02	**5.93E+01**
	rank	3	2	6	4	7	11	9	8	10	5	1
F26	Mean	1.56E+04	1.48E+04	1.50E+04	1.60E+04	1.83E+04	6.41E+04	4.36E+04	2.39E+04	5.15E+04	2.32E+04	**7.11E+03**
	Std	2.64E+03	3.63E+03	1.42E+03	2.14E+03	5.77E+03	7.68E+03	3.29E+03	3.60E+03	2.56E+03	4.74E+03	**4.99E+02**
	rank	4	2	3	5	6	11	9	8	10	7	1
F27	Mean	3.87E+03	3.87E+03	3.88E+03	3.84E+03	4.15E+03	1.32E+04	1.21E+04	4.23E+03	1.45E+04	5.62E+03	**3.47E+03**
	Std	1.42E+02	1.70E+02	1.13E+02	1.82E+02	2.53E+02	2.33E+03	1.18E+03	2.88E+02	1.47E+03	1.06E+03	**4.50E+01**
	rank	4	3	5	2	6	10	9	7	11	8	1
F28	Mean	3.70E+03	3.63E+03	4.01E+03	4.05E+03	4.29E+03	6.20E+04	2.76E+04	1.89E+04	2.99E+04	4.21E+03	**3.45E+03**
	Std	1.12E+02	5.55E+01	2.68E+02	2.63E+02	3.37E+02	1.07E+04	2.52E+03	5.42E+03	1.52E+03	2.33E+02	**3.25E+01**
	rank	3	2	4	5	7	11	9	8	10	6	1
F29	Mean	9.97E+03	9.17E+03	1.00E+04	8.25E+03	1.05E+04	4.06E+06	6.30E+04	1.03E+04	5.69E+05	9.65E+03	**6.14E+03**
	Std	1.01E+03	6.88E+02	8.50E+02	9.69E+02	1.21E+03	4.73E+06	4.64E+04	1.75E+03	3.09E+05	7.97E+02	**4.40E+02**
	rank	5	3	6	2	8	11	9	7	10	4	1
F30	Mean	1.79E+08	1.23E+08	2.40E+08	**4.09E+05**	2.91E+08	6.02E+10	2.27E+10	7.13E+07	3.94E+10	5.78E+07	1.67E+06
	Std	8.27E+07	5.08E+07	1.13E+08	**3.51E+05**	1.27E+08	1.96E+10	4.75E+09	6.42E+07	5.90E+09	2.87E+07	9.69E+05
	rank	6	5	7	1	8	11	9	4	10	3	2
Rank		0	0	0	4	0	0	0	0	0	0	25
Ave. Rank		4.07	2.45	5.00	4.31	6.03	10.79	8.93	7.41	9.83	6.03	1.14
Total Rank		3	2	5	4	6	11	9	8	10	7	1

4.3 Stability Comparison

To further evaluate the robustness and stability of PDMSSA under search landscapes with different levels of complexity, this subsection conducts 30 independent runs on the CEC2017 (100-dimensional) and CEC2022 (20-dimensional) benchmark test sets. Based on the resulting statistical data, box plots are employed to provide a quantitative analysis and visual comparison of algorithmic performance. Box plots offer an intuitive representation of the distribution characteristics of experimental results. The median reflects the central tendency of convergence accuracy, while the height

Table 5: Comparison results on the CEC2022 benchmark functions (10D).

Function	Type	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO	PDMSSA
F1	Mean	**3.00E+02**	**3.00E+02**	**3.00E+02**	**3.00E+02**	**3.00E+02**	4.23E+04	8.91E+03	3.10E+02	6.80E+03	4.90E+02	**3.00E+02**
	Std	5.67E-10	5.69E-10	1.40E-09	6.91E-05	**3.17E-14**	2.15E+04	4.44E+03	3.21E+01	2.24E+03	2.44E+02	4.74E-10
	rank	3	4	5	6	2	11	10	7	9	8	1
F2	Mean	4.06E+02	4.05E+02	4.09E+02	4.03E+02	4.16E+02	1.92E+03	6.88E+02	4.18E+02	1.24E+03	4.20E+02	**4.01E+02**
	Std	1.27E+01	3.83E+00	9.40E+00	**1.64E+00**	2.56E+01	1.02E+03	1.91E+02	2.40E+01	4.82E+02	3.04E+01	3.55E+00
	rank	4	3	5	2	6	11	9	7	10	8	1
F3	Mean	6.12E+02	**6.00E+02**	6.06E+02	**6.00E+02**	6.20E+02	6.75E+02	6.26E+02	6.03E+02	6.46E+02	6.45E+02	**6.00E+02**
	Std	8.69E+00	5.15E-01	7.60E+00	1.86E-03	1.01E+01	1.84E+01	9.34E+00	3.24E+00	6.82E+00	1.14E+01	**1.10E-05**
	rank	6	3	5	2	7	11	8	4	10	9	1
F4	Mean	8.20E+02	8.11E+02	8.20E+02	8.17E+02	8.25E+02	8.96E+02	8.28E+02	8.27E+02	8.48E+02	8.31E+02	**8.08E+02**
	Std	7.85E+00	5.40E+00	6.08E+00	7.06E+00	8.73E+00	2.56E+01	1.02E+01	8.18E+00	7.42E+00	**2.90E+00**	3.38E+00
	rank	5	2	4	3	6	11	8	7	10	9	1
F5	Mean	9.03E+02	**9.00E+02**	9.26E+02	**9.00E+02**	9.72E+02	3.63E+03	1.12E+03	9.28E+02	1.36E+03	1.61E+03	**9.00E+02**
	Std	6.20E+00	1.65E-01	5.16E+01	3.96E-01	1.07E+02	1.02E+03	1.70E+02	3.35E+01	1.85E+02	1.50E+02	**7.02E-11**
	rank	4	2	5	3	7	11	8	6	9	10	1
F6	Mean	4.13E+03	2.01E+03	4.71E+03	**1.81E+03**	3.96E+03	3.66E+08	1.66E+07	4.69E+03	6.66E+06	3.42E+03	2.18E+03
	Std	1.93E+03	1.42E+02	2.12E+03	**1.41E+01**	1.61E+03	4.36E+08	4.13E+07	2.18E+03	9.13E+06	1.98E+03	3.58E+02
	rank	6	2	8	1	5	11	10	7	9	4	3
F7	Mean	2.04E+03	2.03E+03	2.04E+03	**2.01E+03**	2.04E+03	2.17E+03	2.06E+03	2.03E+03	2.08E+03	2.14E+03	**2.01E+03**
	Std	1.59E+01	8.63E+00	1.33E+01	**8.49E+00**	1.48E+01	5.12E+01	2.15E+01	8.96E+00	1.32E+01	6.20E+01	1.01E+01
	rank	5	4	6	1	7	11	8	3	9	10	2
F8	Mean	2.22E+03	2.22E+03	2.23E+03	**2.21E+03**	2.23E+03	2.34E+03	2.27E+03	2.23E+03	2.24E+03	2.27E+03	2.22E+03
	Std	4.98E+00	**3.98E+00**	6.58E+00	8.68E+00	7.87E+00	1.13E+02	5.56E+01	9.05E+00	9.03E+00	6.75E+01	8.77E+00
	rank	4	3	7	1	5	11	9	6	8	10	2
F9	Mean	2.53E+03	2.53E+03	2.54E+03	2.53E+03	2.54E+03	2.85E+03	2.69E+03	2.54E+03	2.73E+03	2.56E+03	**2.52E+03**
	Std	2.68E+01	2.68E+01	2.83E+01	2.92E+01	2.66E+01	1.27E+02	3.30E+01	3.36E+01	2.97E+01	4.92E+01	**6.08E-06**
	rank	4	3	7	2	5	11	9	6	10	8	1
F10	Mean	**2.50E+03**	2.51E+03	2.51E+03	**2.50E+03**	**2.50E+03**	3.16E+03	2.64E+03	2.52E+03	2.68E+03	2.73E+03	**2.50E+03**
	Std	1.44E-01	3.37E+01	3.79E+01	1.37E+01	2.13E-01	6.55E+02	2.54E+02	4.98E+01	1.54E+02	3.84E+02	**5.19E-02**
	rank	3	5	6	2	4	11	8	7	9	10	1
F11	Mean	2.82E+03	2.77E+03	2.78E+03	2.70E+03	2.78E+03	4.94E+04	4.40E+03	2.80E+03	3.49E+03	2.83E+03	**2.67E+03**
	Std	3.38E+02	1.55E+02	**1.23E+02**	1.37E+02	2.21E+02	2.50E+04	7.74E+02	2.72E+02	4.58E+02	3.17E+02	1.49E+02
	rank	7	3	5	2	4	11	10	6	9	8	1
F12	Mean	**2.86E+03**	**2.86E+03**	**2.86E+03**	**2.86E+03**	**2.86E+03**	2.90E+03	3.01E+03	2.87E+03	2.93E+03	2.90E+03	**2.86E+03**
	Std	1.98E+00	1.41E+00	1.40E+00	1.35E+00	1.66E+00	2.30E+01	5.17E+01	5.74E+00	3.69E+01	3.49E+01	**1.19E+00**
	rank	2	3	5	1	4	8	11	7	10	9	6
Global Statistics		0	0	0	4	0	0	0	0	0	0	8
Ave Rank		4.42	3.08	5.33	2.17	5.08	10.75	9.00	6.08	9.33	8.58	1.75
Total Rank		4	3	6	2	5	11	9	7	10	8	1

Table 6: Comparison results on the CEC2022 benchmark functions (20D).

Function	Type	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO	PDMSSA
F1	Mean	3.07E+02	**3.00E+02**	3.83E+03	3.28E+03	1.26E+03	1.32E+05	3.54E+04	1.52E+04	4.06E+04	2.43E+04	**3.00E+02**
	Std	3.08E+01	1.09E-08	2.65E+03	1.94E+03	6.48E+02	4.49E+04	6.82E+03	9.31E+03	1.03E+04	1.17E+04	**5.30E-09**
	rank	3	2	6	5	4	11	9	7	10	8	1
F2	Mean	4.56E+02	4.53E+02	4.55E+02	**4.42E+02**	4.65E+02	4.52E+03	1.10E+03	4.84E+02	2.52E+03	4.61E+02	4.51E+02
	Std	2.36E+01	1.15E+01	2.16E+01	1.71E+01	2.09E+01	1.95E+03	2.57E+02	5.36E+01	6.47E+02	1.63E+01	**5.84E+00**
	rank	5	3	4	1	7	11	9	8	10	6	2
F3	Mean	6.33E+02	6.04E+02	6.22E+02	6.01E+02	6.44E+02	7.08E+02	6.54E+02	6.23E+02	6.77E+02	6.66E+02	**6.00E+02**
	Std	1.48E+01	2.41E+00	1.32E+01	1.15E+00	1.46E+01	2.02E+01	9.49E+00	1.05E+01	6.88E+00	7.93E+00	**1.07E-03**
	rank	6	3	4	2	7	11	8	5	10	9	1
F4	Mean	8.77E+02	8.41E+02	8.65E+02	8.79E+02	8.79E+02	1.08E+03	9.18E+02	9.10E+02	9.65E+02	8.88E+02	**8.16E+02**
	Std	2.47E+01	1.36E+01	1.64E+01	2.14E+01	2.33E+01	3.72E+01	2.01E+01	2.74E+01	1.74E+01	**4.60E+00**	4.62E+00
	rank	4	2	3	6	5	11	9	8	10	7	1
F5	Mean	1.85E+03	9.05E+02	1.65E+03	9.23E+02	2.34E+03	1.12E+04	2.54E+03	1.68E+03	3.45E+03	3.02E+03	**9.00E+02**
	Std	6.10E+02	4.70E+00	5.22E+02	2.59E+01	3.85E+02	2.65E+03	5.99E+02	3.88E+02	3.01E+02	6.02E+02	**9.33E-10**
	rank	6	2	4	3	7	11	8	5	10	9	1
F6	Mean	7.01E+03	3.38E+03	1.64E+04	**2.30E+03**	7.94E+03	2.96E+09	2.71E+08	6.18E+04	1.78E+09	4.41E+03	4.08E+03
	Std	5.42E+03	2.24E+03	8.41E+03	**1.30E+03**	7.28E+03	2.13E+09	2.79E+08	2.70E+05	8.93E+08	4.68E+03	1.85E+03
	rank	5	2	7	1	6	11	9	8	10	4	3
F7	Mean	2.10E+03	2.06E+03	2.12E+03	2.04E+03	2.11E+03	2.38E+03	2.16E+03	2.12E+03	2.19E+03	2.28E+03	**2.03E+03**
	Std	3.03E+01	1.68E+01	4.62E+01	1.58E+01	2.93E+01	9.32E+01	5.90E+01	4.89E+01	3.96E+01	1.33E+02	**5.33E+00**
	rank	4	3	6	2	5	11	8	7	9	10	1
F8	Mean	2.27E+03	2.23E+03	2.31E+03	2.23E+03	2.32E+03	4.06E+03	2.39E+03	2.29E+03	2.42E+03	2.50E+03	**2.22E+03**
	Std	7.17E+01	3.79E+00	6.70E+01	4.88E+00	8.86E+01	3.64E+03	1.51E+02	7.71E+01	1.69E+02	1.66E+02	**1.33E+00**
	rank	4	2	6	3	7	11	8	5	9	10	1
F9	Mean	2.50E+03	**2.48E+03**	2.49E+03	**2.48E+03**	2.50E+03	3.41E+03	2.88E+03	2.49E+03	3.35E+03	2.50E+03	**2.48E+03**
	Std	4.97E+01	1.05E+01	1.80E+01	**5.40E-06**	2.96E+01	5.29E+02	1.43E+02	1.51E+01	3.79E+02	2.42E+01	3.58E-04
	rank	7	3	5	1	8	11	9	4	10	6	2
F10	Mean	3.68E+03	2.89E+03	4.10E+03	**2.51E+03**	3.33E+03	7.23E+03	4.21E+03	2.92E+03	6.34E+03	5.65E+03	2.52E+03
	Std	1.24E+03	7.30E+02	1.35E+03	**3.61E+01**	1.11E+03	9.76E+02	1.26E+03	7.56E+02	1.04E+03	1.71E+03	4.83E+01
	rank	6	3	7	1	5	11	8	4	10	9	2
F11	Mean	**2.91E+03**	**2.90E+03**	3.09E+03	**2.90E+03**	3.00E+03	1.50E+05	1.31E+04	**2.91E+03**	8.62E+03	3.10E+03	**2.90E+03**
	Std	7.22E+01	1.17E-03	1.90E+02	5.49E-01	1.49E+02	6.71E+04	3.03E+03	2.54E+01	6.85E+02	6.35E+02	**4.75E-04**
	rank	5	2	7	3	6	11	10	4	9	8	1
F12	Mean	2.96E+03	2.96E+03	2.97E+03	**2.95E+03**	2.98E+03	3.31E+03	3.64E+03	2.99E+03	3.49E+03	3.24E+03	**2.95E+03**
	Std	2.04E+01	2.07E+01	1.63E+01	7.23E+00	2.58E+01	1.61E+02	1.88E+02	3.60E+01	1.96E+02	1.50E+02	**6.96E+00**
	rank	4	3	5	2	6	9	11	7	10	8	1
Rank		0	0	0	4	0	0	0	0	0	0	8
Ave Rank		4.92	2.50	5.33	2.50	6.08	10.83	8.83	6.00	9.75	7.83	1.42
Total Rank		4	2	5	3	7	11	9	6	10	8	1

of the box, defined by the interquartile range (IQR), characterizes the dispersion of the core solution set. A narrower IQR indicates a more concentrated distribution and, consequently, higher performance stability. The upper and lower whiskers delimit the range of non-outlier values, whereas outliers correspond to results that deviate from the main distribution in individual runs, reflecting performance fluctuations under different stochastic conditions.

Fig. 7 shows box plots of 30 independent runs on nine representative 100-dimensional CEC2017 functions (F1, F5, F8, F9, F10, F16, F20, F21, and F29),

covering unimodal, multimodal, hybrid, and composition problem types. The box plots are used to compare the distribution characteristics of different algorithms under complex high-dimensional search conditions. Across all selected functions, PDMSSA consistently exhibits more compact box distributions than most comparison algorithms, characterized by smaller interquartile ranges and shorter whiskers. For the unimodal function F1 and the multimodal functions F5 and F9, PDMSSA shows a lower median fitness and reduced dispersion, whereas several competing algorithms present wider boxes and more extreme values, indicating higher variability across runs. In the hybrid functions F16 and F20, although most algorithms experience increased dispersion due to strong variable coupling, the distribution of PDMSSA remains comparatively concentrated with a competitive median. A similar pattern is observed for the composition functions F21 and F29, where PDMSSA demonstrates a narrower distribution and fewer outliers than most competitors. Overall, the box plot analysis indicates that PDMSSA yields more stable solution distributions than the compared algorithms across different categories of high-dimensional test functions, with consistently reduced performance variability over multiple runs.

Fig. 8 presents box plots obtained from 30 independent runs on six representative CEC2022 functions under the 20-dimensional setting (F1, F3, F4, F5, F8, and F12). These box plots are used to compare the performance distributions of different algorithms in a moderate-dimensional search space. Across all selected functions, PDMSSA generally exhibits more compact box distributions than most competing algorithms, characterized by smaller interquartile ranges and shorter whiskers. For the unimodal function F1 and the basic functions F3 and F5, PDMSSA shows lower or comparable median values with reduced dispersion, whereas some algorithms, such as HSSAPSO and MSNSSA, present wider box ranges and higher variability across runs. In the multimodal function F4, several competing algorithms display wider distributions with longer whiskers and a larger number of outliers, while the results of PDMSSA remain comparatively concentrated. For the structurally more complex hybrid function F8 and the composition function F12, PDMSSA maintains lower median values and more compact box ranges than some comparison algorithms (e.g., MSNSSA and CPO), indicating reduced performance variability. Overall, the box plot analysis in Figure 8 indicates that PDMSSA yields more concentrated result distributions and lower variability than the comparison algorithms under the CEC2022 20-dimensional setting, reflecting stable performance across different function types.

Overall, the box plot results show that PDMSSA produces consistently concentrated performance distributions across both low- and high-dimensional test settings. Across different categories of benchmark functions, the interquartile ranges remain relatively small and the median values vary within a narrow range, indicating stable convergence behavior under diverse search landscapes. In addition, the limited whisker lengths and the small number of extreme values suggest that the variability across independent runs is well controlled. These observations support that PDMSSA maintains stable optimization behavior when applied to complex multidimensional problems.

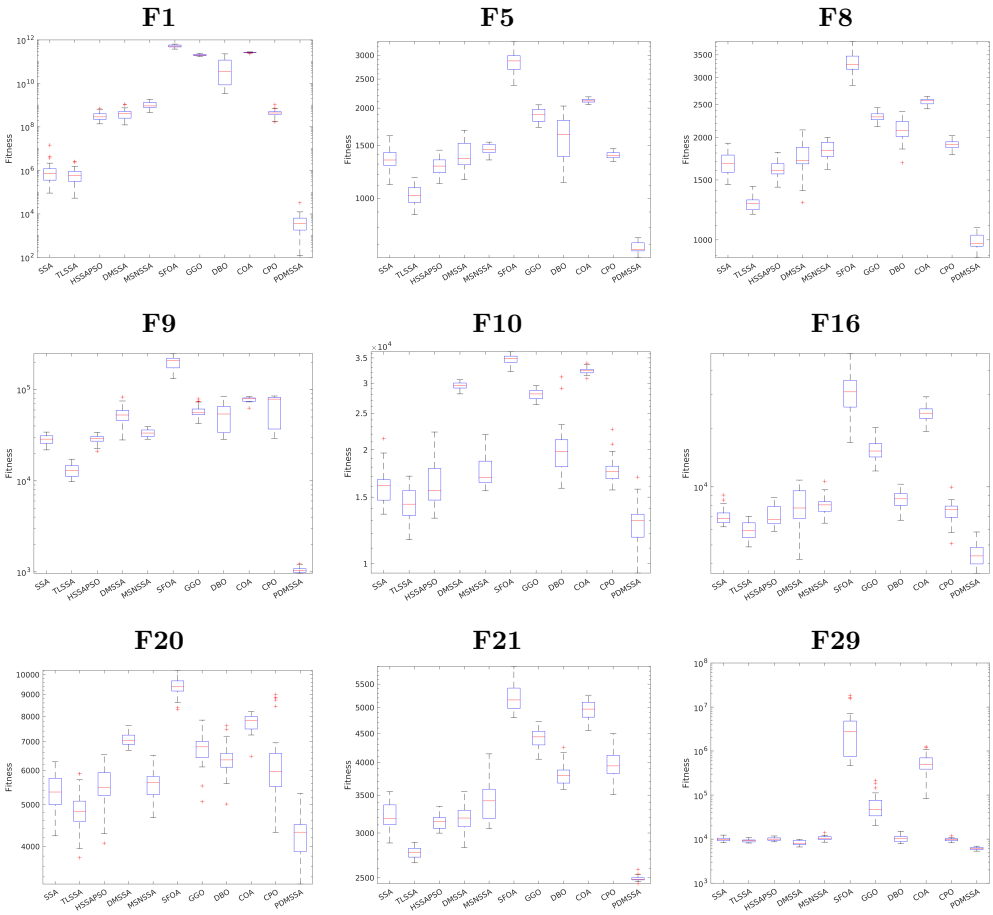


Fig. 7: Boxplots of different algorithms on the CEC2017 benchmark functions (100D).

4.4 Comparisons of convergence performance

To examine the search dynamics of PDMSSA and its convergence behavior under different optimization landscapes, a comparative analysis is conducted using the CEC2017 and CEC2022 benchmark suites. For each test function, all algorithms are independently executed 30 times under identical parameter settings, and the mean evolution of the best fitness value over iterations is recorded to depict the convergence process.As the selected benchmark suites include unimodal, multimodal, hybrid, and composition functions with varying landscape characteristics, representative functions from each category are chosen for convergence comparison. This selection allows the main convergence patterns of different algorithms to be illustrated without introducing redundant visual information.

Fig. 9 illustrates the average convergence curves of all algorithms on representative CEC2017 test functions under the 100-dimensional setting. Clear differences in

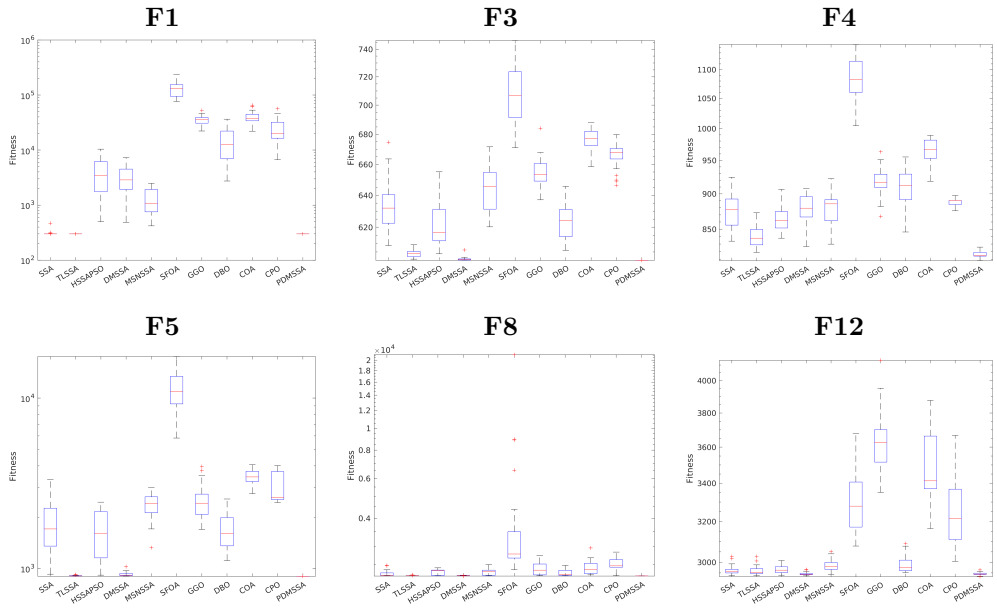


Fig. 8: Boxplots of different algorithms on the CEC2022 benchmark functions(20D).

convergence behavior can be observed across different function categories and evolutionary stages. For the unimodal function F1 and the simple multimodal functions F5 and F9, PDMSSA shows a rapid reduction in fitness values during the early iterations, indicating efficient initial convergence in high-dimensional search spaces. In comparison, several competing algorithms exhibit slower fitness improvement in the same stage. For hybrid functions with strong variable coupling, such as F12 and F18, as well as the structurally complex function F26, many algorithms experience evident stagnation during the evolutionary process. By contrast, PDMSSA maintains a steady decrease in fitness values in the middle and later stages, leading to improved final solutions. For multimodal hybrid functions such as F10 and F20, PDMSSA exhibits a relatively slower fitness reduction in the early iterations compared with some competitors. However, its convergence curves display a more consistent downward trend in the subsequent stages, eventually achieving better final fitness values. Overall, the convergence results demonstrate that PDMSSA maintains stable convergence behavior across different function types under high-dimensional conditions and achieves competitive final performance on multiple challenging test functions.

Fig. 10 presents the average convergence behaviors of representative CEC2022 functions under the 20-dimensional setting, including unimodal, multimodal, hybrid, and composition landscapes. For the unimodal function F1, PDMSSA shows a rapid fitness reduction in the early iterations and converges stably thereafter, while MSNSSA and DMSSA exhibit slower decrease in the same stage. For the basic multimodal functions F3 and F4, PDMSSA maintains a sustained downward trend throughout the

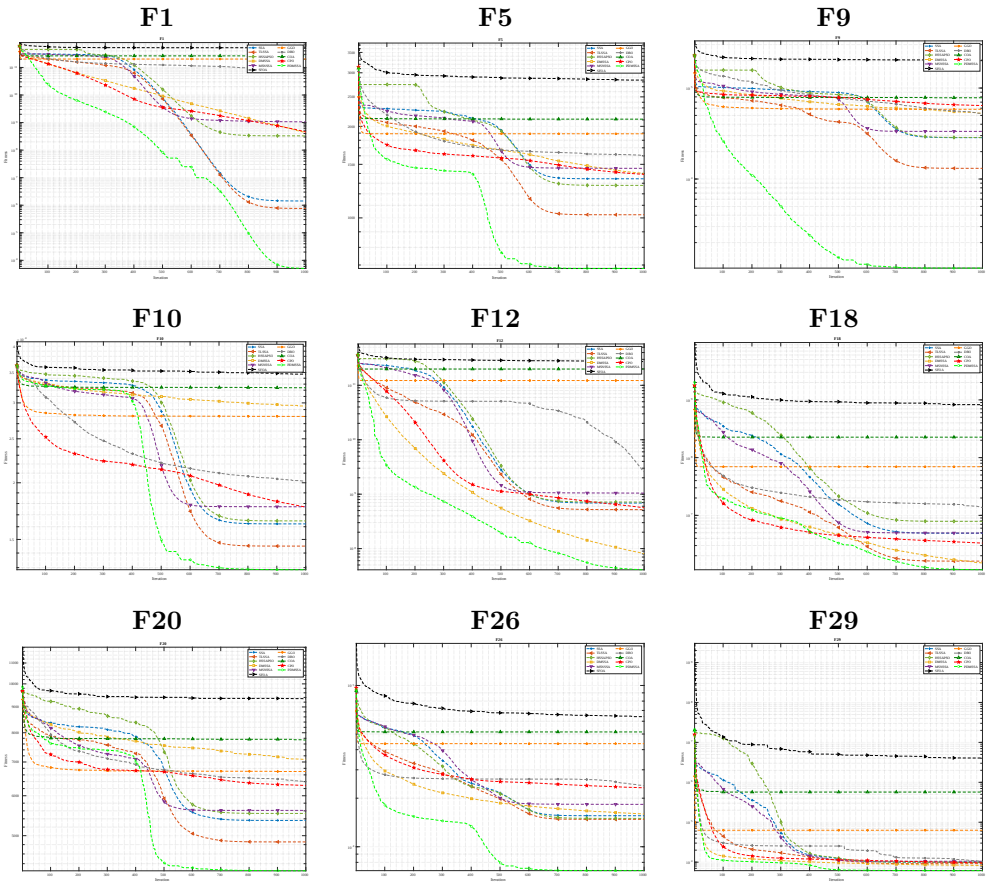


Fig. 9: Convergence curves of different algorithms on the CEC2017 benchmark functions (100D).

evolutionary process, whereas algorithms such as SSA and GGO experience convergence slowdown in the middle or later stages. In the hybrid function F7, PDMSSA exhibits convergence behavior comparable to TLSSA in the early stage and gradually attains lower fitness values as iterations proceed. For the composition functions F10 and F12, although some algorithms achieve faster initial reduction, they tend to stagnate earlier, while PDMSSA continues improving solution quality in the middle and later stages and achieves better final results. Overall, the convergence curves indicate that PDMSSA maintains stable and sustained solution improvement across different function types under the 20-dimensional CEC2022 setting.

Overall, PDMSSA demonstrates stable and competitive convergence performance under both the low-dimensional CEC2022 and the high-dimensional CEC2017 test settings.

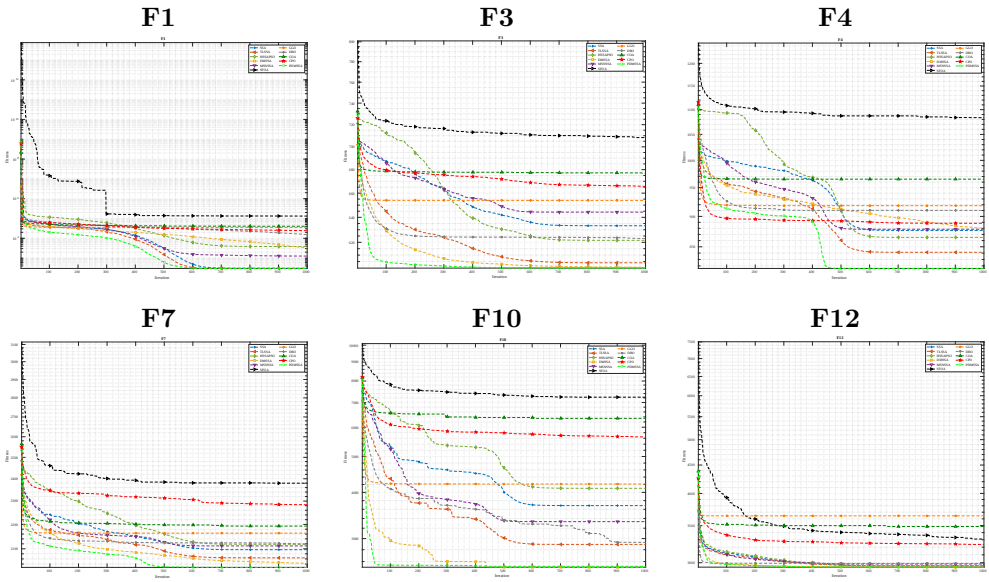


Fig. 10: Convergence curves of different algorithms on the CEC2022 benchmark functions (20D).

4.5 Effectiveness Analysis of Individual Strategies

To investigate the individual contributions of the proposed improvement strategies in PDMSSA and to examine their combined effects, an ablation study is conducted based on the CEC2017 benchmark suite. By decoupling the core components of PDMSSA, three variant algorithms are constructed to evaluate the specific role of each mechanism in the optimization process. Specifically, SSA1 introduces a fast non-dominated sorting–based leader selection mechanism into the original salp swarm algorithm (SSA) to improve leader selection. SSA2 incorporates a scale-free network–based multi-source information fusion strategy to enhance information exchange among population individuals. SSA3 integrates a state-aware directional tabu search strategy to strengthen local improvement capability during stagnation phases. All variant algorithms and the complete PDMSSA are independently executed 30 times under identical parameter settings. Their performances are compared in terms of convergence accuracy and related statistical indicators to assess the impact of each individual strategy. As shown by the ablation results in Table 7, all three variants that incorporate a single improvement strategy outperform the original SSA in terms of convergence accuracy and stability. This indicates that each proposed mechanism contributes positively to optimization performance when applied independently. Moreover, PDMSSA, which integrates all three strategies, achieves further performance improvement and attains the best overall mean rank of 1.14 among all compared algorithms. Compared with the single-strategy variants, PDMSSA produces superior or more stable results on the majority of test functions, reflecting the complementary effects of the combined strategies.

Overall, the ablation results confirm that integrating multiple improvement modules within a unified framework leads to enhanced optimization performance, improving both convergence accuracy and stability on complex benchmark problems.

Table 7: Ablation study results under different strategies.

Function	Type	SSA	SSA1	SSA2	SSA3	PDMSSA
F1	Mean	9.54E+05	2.16E+05	7.39E+04	6.53E+03	**5.59E+03**
	Std	8.30E+05	1.24E+06	6.66E+04	**4.94E+03**	4.96E+03
	rank	5	4	3	2	1
F3	Mean	4.67E+05	4.20E+05	4.39E+05	**2.24E+05**	2.70E+05
	Std	1.47E+05	1.18E+05	5.81E+04	**3.38E+04**	4.42E+04
	rank	5	3	4	1	2
F4	Mean	9.70E+02	9.07E+02	8.03E+02	7.25E+02	**7.11E+02**
	Std	9.18E+01	5.95E+01	4.45E+01	3.37E+01	**2.73E+01**
	rank	5	4	3	2	1
F5	Mean	1.34E+03	**1.20E+02**	1.10E+03	7.85E+02	6.69E+02
	Std	7.53E+01	1.13E+02	1.66E+02	4.12E+01	**3.38E+01**
	rank	5	1	4	3	2
F6	Mean	6.73E+02	6.71E+02	6.50E+02	6.05E+02	**6.02E+02**
	Std	4.07E+00	4.66E+00	9.90E+00	1.39E+00	**5.89E-01**
	rank	5	4	3	2	1
F7	Mean	2.26E+03	2.23E+03	1.99E+03	1.06E+03	**9.93E+02**
	Std	2.65E+02	2.26E+02	1.65E+02	7.79E+01	**3.77E+01**
	rank	5	4	3	2	1
F8	Mean	1.74E+03	1.74E+03	1.35E+03	1.05E+03	**9.75E+02**
	Std	1.71E+02	1.65E+02	1.03E+02	**3.17E+01**	4.77E+01
	rank	5	4	3	2	1
F9	Mean	3.06E+04	3.03E+04	2.13E+04	4.80E+03	**1.03E+03**
	Std	3.54E+03	2.64E+03	5.28E+03	2.26E+03	**5.64E+01**
	rank	5	4	3	2	1
F10	Mean	1.70E+04	1.56E+04	1.33E+04	1.34E+04	**1.28E+04**
	Std	1.49E+03	1.24E+03	1.60E+03	1.30E+03	**8.40E+02**
	rank	5	4	2	3	1
F11	Mean	3.00E+04	2.64E+04	6.26E+03	5.87E+03	**3.25E+03**
	Std	1.60E+04	1.08E+04	1.73E+03	1.59E+03	**2.06E+02**
	rank	5	4	3	2	1
F12	Mean	7.07E+08	5.32E+08	2.08E+08	4.75E+07	**4.32E+07**
	Std	2.18E+08	1.80E+08	8.83E+07	**1.42E+07**	2.22E+07
	rank	5	4	3	2	1
F13	Mean	8.94E+04	7.32E+04	2.98E+04	3.50E+04	**2.95E+04**
	Std	2.88E+04	3.01E+04	4.32E+05	**1.20E+04**	1.29E+04
	rank	5	4	2	3	1
F14	Mean	2.68E+06	2.66E+06	1.75E+06	5.32E+05	**4.98E+05**
	Std	8.76E+05	1.86E+06	1.15E+06	**1.48E+05**	2.04E+05
	rank	5	4	3	2	1
F15	Mean	8.44E+04	4.82E+04	8.07E+04	2.60E+04	**2.49E+04**
	Std	4.40E+04	2.22E+04	3.23E+04	1.13E+04	**6.43E+03**
	rank	5	3	4	2	1
F16	Mean	7.03E+03	6.19E+03	6.05E+03	4.63E+03	**4.30E+03**
	Std	6.18E+02	7.34E+02	9.72E+02	1.06E+03	**5.48E+02**
	rank	5	4	3	2	1
F17	Mean	5.90E+03	5.54E+03	5.25E+03	4.39E+03	**3.79E+03**
	Std	5.33E+02	**3.85E+02**	6.57E+02	5.87E+02	4.13E+02
	rank	5	4	3	2	1
F18	Mean	5.59E+06	3.15E+06	3.49E+06	9.90E+05	**9.16E+05**
	Std	3.20E+06	1.91E+06	1.70E+06	**2.64E+05**	4.01E+05
	rank	5	3	4	2	1
F19	Mean	1.66E+07	1.45E+07	6.23E+05	4.66E+04	**3.44E+04**
	Std	1.14E+07	8.22E+06	4.89E+05	2.55E+04	**1.68E+04**
	rank	5	4	3	2	1
F20	Mean	5.59E+03	5.24E+03	**4.09E+03**	4.34E+03	4.18E+03
	Std	7.10E+02	4.94E+02	5.77E+02	5.19E+02	**4.61E+02**
	rank	5	4	1	3	2
F21	Mean	3.19E+03	3.16E+03	2.92E+03	**2.51E+03**	**2.51E+03**
	Std	1.13E+02	1.49E+02	1.27E+02	3.26E+01	**3.04E+01**
	rank	5	4	3	2	1
F22	Mean	1.80E+04	2.03E+04	1.75E+04	1.54E+04	**1.08E+04**
	Std	5.78E+03	1.61E+03	**1.35E+03**	7.44E+03	1.72E+03
	rank	4	5	3	2	1
F23	Mean	3.63E+03	3.61E+03	3.53E+03	3.03E+03	**3.00E+03**
	Std	8.48E+01	1.38E+02	1.84E+02	3.42E+01	**3.31E+01**
	rank	5	4	3	2	1
F24	Mean	4.21E+03	4.20E+03	3.98E+03	3.47E+03	**3.42E+03**
	Std	2.05E+02	1.87E+02	2.06E+02	4.09E+01	**3.96E+01**
	rank	5	4	3	2	1
F25	Mean	3.64E+03	3.63E+03	3.52E+03	3.39E+03	**3.34E+03**
	Std	1.01E+02	6.85E+01	5.62E+01	4.96E+01	**4.72E+01**
	rank	5	4	3	2	1
F26	Mean	1.53E+04	1.51E+04	1.27E+04	**6.16E+03**	7.29E+03
	Std	1.85E+03	1.42E+03	1.83E+03	2.88E+03	**5.91E+02**
	rank	5	4	3	1	2
F27	Mean	3.90E+03	3.84E+03	3.74E+03	3.52E+03	**3.49E+03**
	Std	1.95E+02	1.02E+02	1.91E+02	**3.90E+01**	4.68E+01
	rank	5	4	3	2	1
F28	Mean	3.72E+03	3.71E+03	3.51E+03	3.52E+03	**3.47E+03**
	Std	9.39E+01	9.51E+01	9.05E+01	**3.33E+01**	3.52E+01
	rank	5	4	2	3	1
F29	Mean	9.69E+03	9.35E+03	8.67E+03	6.35E+03	**6.08E+03**
	Std	1.07E+03	1.05E+03	7.43E+02	**4.24E+02**	6.16E+02
	rank	5	4	3	2	1
F30	Mean	1.63E+08	1.33E+08	1.40E+07	2.99E+06	**1.69E+06**
	Std	9.63E+07	4.67E+07	8.75E+06	1.18E+06	**8.20E+05**
	rank	5	4	3	2	1
Count (First Place)		0	1	1	2	25
Ave Rank		4.97	3.83	2.97	2.10	1.14
Total Rank		5	4	3	2	1

4.6 Nonparametric Statistical Tests

4.6.1 Friedman test

To statistically examine whether the observed performance differences among the compared algorithms are significant, the Friedman test and its improved variant, the Iman–Davenport test, are employed. As rank-based nonparametric methods, these tests do not require assumptions regarding the normality of performance distributions and are therefore suitable for comparative analysis in evolutionary computation studies.

Assuming N test functions and k competing algorithms, let r_i^j denote the rank of the j-th algorithm on the i-th test function. The average rank of each algorithm is computed as Eq. 20.

$$R_j = \frac{1}{N} \sum_{i=1}^{N} r_i^j \tag{20}$$

Under the null hypothesis H_0, all algorithms exhibit equivalent performance and thus have identical average ranks. The Friedman statistic χ_F^2, defined in Eq. 21, approximately follows a chi-square distribution with $k-1$ degrees of freedom. To improve accuracy under finite-sample conditions, the Iman–Davenport statistic F_F, given in Eq. 22, is further adopted, which follows an F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom.

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{21}$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} \tag{22}$$

The Friedman and Iman–Davenport tests are applied to evaluate the performance differences among PDMSSA and the remaining ten competing algorithms ($k = 11$) on the CEC2017 and CEC2022 benchmark test suites. As reported in Table 8 and Table 9, all obtained p-values under different dimensional settings are far below the significance level $\alpha = 0.05$. Specifically, for CEC2017 under 50D and 100D, the Friedman test yields p-values of 1.32×10^{-46} and 7.13×10^{-105}, while the corresponding Iman–Davenport test produces p-values of 2.35×10^{-46} and 2.01×10^{-103}. For CEC2022 under 10D and 20D, the Friedman test yields p-values of 7.01×10^{-17} and 3.27×10^{-35}, and the Iman–Davenport test yields p-values of 6.31×10^{-18} and 4.39×10^{-44}.

Table 8: Friedman and Iman–Davenport test results on CEC2017.

Dimension	P_{Friedman}	P_{Iman}	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO	PDMSSA
50	1.32×10^{-46}	7.13×10^{-105}	4.21	2.72	5.14	3.48	5.69	10.79	8.97	7.07	9.93	6.72	1.28
100	1.87×10^{-46}	2.01×10^{-103}	4.07	2.45	5.00	4.31	6.03	10.79	8.93	7.41	9.83	6.03	1.14

These results lead to a rejection of the null hypothesis, indicating statistically significant performance differences among the compared algorithms. Further inspection

Table 9: Friedman and Iman–Davenport test results on CEC2022.

Dimension	P_{Friedman}	P_{Iman}	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO	PDMSSA
10	7.0067×10^{-17}	3.27×10^{-35}	4.42	3.08	5.33	2.17	5.08	10.75	9.00	6.08	9.33	8.58	1.75
20	6.3189×10^{-18}	4.39×10^{-44}	4.92	2.50	5.33	2.50	6.08	10.83	8.83	6.00	9.75	7.83	1.42

of the average ranks shows that PDMSSA consistently attains the lowest mean rank across all benchmark settings, including CEC2017 (50D and 100D) and CEC2022 (10D and 20D). Under the established statistical significance, this ranking consistency reflects the competitive overall performance of PDMSSA across different dimensionalities and benchmark categories.

4.6.2 Wilcoxon rank-sum test

To complement the overall multi-algorithm comparison based on the Friedman test, the Wilcoxon rank-sum test is employed to perform pairwise statistical comparisons between PDMSSA and each competing algorithm. As a nonparametric test for two independent samples, the Wilcoxon rank-sum test does not require normality assumptions and is therefore suitable for analyzing the stochastic outcomes of metaheuristic optimization algorithms. For each benchmark setting, the null hypothesis H_0 assumes that there is no statistically significant difference between the performance distributions of PDMSSA and a given competing algorithm at the significance level $\alpha = 0.05$. Each comparison is conducted using two independent samples of size $n_1 = n_2 = 30$, corresponding to the objective function values obtained over repeated runs. The Mann–Whitney statistic U and its standardized form Z are computed according to Eq. 23 and Eq. 24. When the resulting p-value is smaller than α, the null hypothesis is rejected.

$$U = R_1 - \frac{n_1(n_1+1)}{2} \tag{23}$$

$$Z = \frac{U - \frac{n_1 n_2}{2}}{\sqrt{\frac{n_1 n_2 (n_1+n_2+1)}{12}}} \tag{24}$$

For concise presentation, the outcomes of the Wilcoxon tests are summarized using the symbols "+", "−", and "=", indicating that PDMSSA performs significantly better than, worse than, or statistically indistinguishable from the compared algorithm, respectively.

Table 10 to Table 13 report the Wilcoxon rank-sum test results for the CEC2017 (50D and 100D) and CEC2022 (10D and 20D) benchmark suites. The results show that the "+" outcome dominates across most pairwise comparisons, indicating that PDMSSA achieves statistically better performance than the competing algorithms in the majority of test cases. In particular, PDMSSA maintains consistent statistical advantages under the high-dimensional settings of CEC2017, while also exhibiting stable performance in the low- and moderate-dimensional scenarios of CEC2022. The occurrences of "−" and "=" are relatively limited, suggesting that PDMSSA rarely underperforms or becomes statistically indistinguishable from the comparison methods.

Table 10: Wilcoxon rank-sum test results on the CEC2017 benchmark functions (50D)

Function	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO
F1	0.8187(=)	0.0010(-)	0.3329(=)	3.02e-11(+)	1.56e-08(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F3	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F4	0.0053(+)	0.0271(+)	1.02e-05(+)	0.7172(=)	2.78e-07(+)	3.02e-11(+)	3.02e-11(+)	1.96e-10(+)	3.02e-11(+)	4.69e-08(+)
F5	3.02e-11(+)	3.69e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F6	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F7	3.02e-11(+)	5.49e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F8	3.02e-11(+)	3.69e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F9	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F10	9.06e-08(+)	0.0004(+)	8.89e-10(+)	3.02e-11(+)	1.01e-08(+)	3.02e-11(+)	3.02e-11(+)	8.15e-11(+)	3.02e-11(+)	3.16e-10(+)
F11	8.99e-11(+)	6.12e-10(+)	3.02e-11(+)	2.49e-06(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	4.98e-11(+)
F12	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	0.4035(=)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F13	8.99e-11(+)	3.20e-09(+)	3.34e-11(+)	4.08e-05(-)	6.07e-11(+)	3.02e-11(+)	3.02e-11(+)	5.49e-11(+)	3.02e-11(+)	2.67e-09(+)
F14	7.09e-08(+)	1.86e-09(+)	3.02e-11(+)	5.97e-09(-)	4.62e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.34e-11(+)
F15	3.02e-11(+)	5.57e-10(+)	3.69e-11(+)	0.0615(=)	1.33e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	8.48e-09(+)
F16	4.98e-11(+)	1.17e-09(+)	3.34e-11(+)	1.21e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F17	3.16e-10(+)	6.53e-08(+)	2.61e-10(+)	1.07e-07(+)	8.99e-11(+)	3.02e-11(+)	3.02e-11(+)	6.07e-11(+)	3.02e-11(+)	8.15e-11(+)
F18	8.99e-11(+)	2.03e-07(+)	9.92e-11(+)	0.0070(+)	2.03e-09(+)	3.02e-11(+)	3.02e-11(+)	6.12e-10(+)	3.02e-11(+)	2.83e-08(+)
F19	3.02e-11(+)	8.15e-11(+)	3.02e-11(+)	0.1055(=)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.34e-11(+)	3.02e-11(+)	1.21e-10(+)
F20	3.82e-09(+)	0.0001(+)	1.70e-08(+)	1.09e-10(+)	3.96e-08(+)	3.02e-11(+)	3.34e-11(+)	2.15e-10(+)	3.02e-11(+)	3.34e-11(+)
F21	3.02e-11(+)	4.50e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F22	1.85e-08(+)	0.5298(=)	1.86e-09(+)	3.02e-11(+)	1.61e-10(+)	3.02e-11(+)	3.02e-11(+)	8.15e-11(+)	3.02e-11(+)	6.70e-11(+)
F23	3.02e-11(+)	6.70e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F24	3.02e-11(+)	6.72e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F25	0.0003(+)	1.29e-09(+)	4.18e-09(+)	6.05e-07(+)	7.38e-10(+)	3.02e-11(+)	3.02e-11(+)	1.33e-10(+)	3.02e-11(+)	3.02e-11(+)
F26	0.1858(=)	0.3790(=)	1.07e-07(+)	5.97e-09(+)	0.0020(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	1.29e-09(+)
F27	3.02e-11(+)	3.34e-11(+)	3.02e-11(+)	9.26e-09(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F28	1.73e-07(+)	3.81e-07(+)	6.07e-11(+)	1.70e-08(+)	2.37e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F29	3.02e-11(+)	4.08e-11(+)	3.02e-11(+)	3.47e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F30	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	1.39e-06(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.69e-11(+)	3.02e-11(+)	3.02e-11(+)
W/T/L	27/2/0	26/2/1	28/1/0	23/4/2	29/0/0	29/0/0	29/0/0	29/0/0	29/0/0	29/0/0

Table 11: Wilcoxon rank-sum test results on the CEC2017 benchmark functions (100D).

Function	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO
F1	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F3	4.20e-10(+)	0.0232(+)	1.78e-10(+)	3.02e-11(+)	3.96e-08(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	4.98e-11(+)	6.07e-11(+)
F4	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F5	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F6	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F7	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F8	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F9	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F10	1.31e-08(+)	0.0004(+)	3.65e-08(+)	3.02e-11(+)	1.78e-10(+)	3.02e-11(+)	3.02e-11(+)	4.08e-11(+)	3.02e-11(+)	8.15e-11(+)
F11	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F12	3.02e-11(+)	3.02e-11(+)	4.08e-11(+)	8.29e-06(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F13	1.21e-10(+)	1.02e-05(+)	4.08e-11(+)	0.5793(=)	8.35e-08(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	7.38e-10(+)
F14	3.69e-11(+)	6.01e-08(+)	3.02e-11(+)	0.0003(-)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	6.07e-11(+)
F15	1.17e-09(+)	9.79e-05(+)	4.98e-11(+)	0.0006(-)	7.38e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	2.43e-05(+)
F16	3.02e-11(+)	1.69e-09(+)	3.02e-11(+)	1.41e-09(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	6.07e-11(+)
F17	3.34e-11(+)	1.78e-10(+)	6.07e-11(+)	1.33e-10(+)	4.08e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	4.08e-11(+)
F18	1.55e-09(+)	0.0014(+)	1.33e-10(+)	0.0824(=)	6.12e-10(+)	3.02e-11(+)	3.02e-11(+)	5.49e-11(+)	3.02e-11(+)	9.26e-09(+)
F19	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.50e-09(-)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F20	5.97e-09(+)	1.75e-05(+)	4.18e-09(+)	3.02e-11(+)	1.96e-10(+)	3.02e-11(+)	4.08e-11(+)	4.08e-11(+)	3.02e-11(+)	3.82e-10(+)
F21	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F22	2.03e-09(+)	5.53e-08(+)	4.08e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F23	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F24	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F25	3.02e-11(+)	4.08e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F26	3.02e-11(+)	6.12e-10(+)	3.02e-11(+)	3.02e-11(+)	1.07e-09(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F27	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.34e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F28	8.15e-11(+)	3.34e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F29	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	8.15e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F30	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	4.18e-09(-)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
W/T/L	29/0/0	29/0/0	29/0/0	23/2/4	29/0/0	29/0/0	29/0/0	29/0/0	29/0/0	29/0/0

Taken together with the Friedman-based ranking analysis, the Wilcoxon test results provide further statistical evidence that the proposed PDMSSA achieves stable and consistent performance advantages across different dimensional settings and benchmark categories.

5 UAV Trajectory Optimization

The three-dimensional trajectory planning problem for unmanned aerial vehicles (UAVs) is characterized by nonlinear search spaces, complex environmental constraints, and high-dimensional decision variables. The objective is to generate a feasible

Table 12: Wilcoxon rank-sum test results on the CEC2022 benchmark functions (10D).

Function	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO
F1	2.20e-07(+)	1.56e-08(+)	9.92e-11(+)	0.0170(+)	1.08e-11(-)	3.02e-11(+)	3.02e-11(+)	0.1842(=)	3.02e-11(+)	3.02e-11(+)
F2	0.5895(=)	0.6204(=)	0.0023(+)	2.13e-05(+)	0.0103(+)	3.02e-11(+)	3.02e-11(+)	5.44e-06(+)	3.02e-11(+)	0.0292(+)
F3	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	2.02e-08(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F4	6.12e-10(+)	0.0046(+)	6.12e-10(+)	5.60e-07(+)	1.07e-09(+)	3.02e-11(+)	1.21e-10(+)	6.69e-11(+)	3.02e-11(+)	3.02e-11(+)
F5	9.91e-11(+)	1.61e-10(+)	3.02e-11(+)	1.17e-05(+)	1.11e-06(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F6	8.84e-07(+)	0.1154(=)	3.35e-08(+)	4.08e-11(-)	5.97e-09(+)	3.02e-11(+)	4.31e-08(+)	3.83e-06(+)	3.02e-11(+)	0.0007(+)
F7	1.43e-08(+)	3.26e-07(+)	2.67e-09(+)	0.0073(-)	3.16e-10(+)	3.02e-11(+)	3.02e-11(+)	3.26e-07(+)	3.02e-11(+)	3.02e-11(+)
F8	1.31e-08(+)	2.44e-09(+)	4.20e-10(+)	0.0392(-)	3.50e-09(+)	3.02e-11(+)	3.02e-11(+)	2.67e-09(+)	3.02e-11(+)	3.34e-11(+)
F9	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	4.11e-12(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	1.0000(=)	3.02e-11(+)	3.02e-11(+)
F10	8.15e-11(+)	1.61e-06(+)	5.49e-11(+)	1.49e-06(+)	2.87e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F11	0.0170(+)	0.0014(-)	0.7958(=)	0.0011(+)	0.0635(=)	3.02e-11(+)	1.78e-10(+)	0.0335(+)	5.00e-09(+)	0.0170(+)
F12	0.0005(-)	2.13e-05(-)	0.1494(=)	4.96e-10(-)	0.5011(=)	2.98e-11(+)	3.02e-11(+)	0.0215(+)	3.02e-11(+)	3.02e-11(+)
W/T/L	10/1/1	8/2/2	10/2/0	8/0/4	9/2/1	12/0/0	12/0/0	10/2/0	12/0/0	12/0/0

Table 13: Wilcoxon rank-sum test results on the CEC2022 benchmark functions (20D).

Function	SSA	TLSSA	HSSAPSO	DMSSA	MSNSSA	SFOA	GGO	DBO	COA	CPO
F1	2.77e-05(+)	0.0007(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F2	0.1297(=)	0.4918(=)	0.0026(+)	0.0056(-)	3.57e-06(+)	3.02e-11(+)	3.02e-11(+)	0.0053(+)	3.02e-11(+)	6.53e-08(+)
F3	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F4	3.02e-11(+)	6.07e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F5	3.02e-11(+)	2.37e-10(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F6	0.0824(=)	0.0594(=)	3.32e-06(+)	3.52e-07(-)	0.2116(=)	3.02e-11(+)	3.02e-11(+)	0.0004(+)	3.02e-11(+)	0.1413(=)
F7	3.02e-11(+)	4.08e-11(+)	4.98e-11(+)	2.15e-06(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	4.98e-11(+)	3.02e-11(+)	3.02e-11(+)
F8	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	1.29e-09(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	5.49e-11(+)	3.02e-11(+)	3.02e-11(+)
F9	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	9.92e-11(-)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)
F10	9.06e-08(+)	2.43e-05(+)	5.00e-09(+)	0.0001(+)	1.73e-07(+)	3.02e-11(+)	5.57e-10(+)	1.11e-06(+)	4.50e-11(+)	5.49e-11(+)
F11	3.02e-11(+)	3.02e-11(+)	3.02e-11(+)	4.50e-11(+)	0.1858(=)	3.02e-11(+)	3.02e-11(+)	7.88e-09(+)	3.02e-11(+)	3.02e-11(+)
F12	1.25e-05(+)	0.0004(+)	3.81e-07(+)	0.7283(=)	2.67e-09(+)	3.02e-11(+)	3.02e-11(+)	1.46e-10(+)	3.02e-11(+)	3.02e-11(+)
W/T/L	10/2/0	10/2/0	12/0/0	8/1/3	10/2/0	12/0/0	12/0/0	12/0/0	12/0/0	11/1/0

flight trajectory with minimal overall cost in a three-dimensional environment containing undulating terrain and discrete threat regions, while satisfying flight feasibility and safety requirements. As environmental complexity and trajectory freedom increase, the associated search space expands rapidly, rendering conventional deterministic or rule-based planning methods increasingly difficult to apply effectively under limited computational resources. Consequently, metaheuristic optimization algorithms have become a widely adopted approach for addressing such complex trajectory optimization problems.

5.1 Environment Modeling and Coordinate System Definition

To represent the flight environment under complex terrain conditions in three-dimensional space, a digital elevation map (DEM), denoted as $H(x, y)$, is employed to describe terrain undulations. The absolute flight altitude of the UAV at the i-th trajectory node, $Z_{\text{abs}}^{(i)}$, is defined as the sum of the terrain elevation and the altitude relative to the ground surface, as given in Eq. 25, where $z^{(i)}$ denotes the relative altitude variable directly optimized by the algorithm. This formulation explicitly couples the UAV flight height with the underlying terrain, allowing terrain effects to be consistently incorporated into the trajectory optimization process. The spatial distribution and elevation characteristics of the terrain in the experimental environment are illustrated in Fig. 11.

$$Z_{\text{abs}}^{(i)} = z^{(i)} + H\left(x^{(i)}, y^{(i)}\right) \tag{25}$$

$$C_{\text{free}} = \left\{ (x, y) \;\middle|\; \forall k \in [1, K],\ \sqrt{(x - x_k)^2 + (y - y_k)^2} > R_k + R_{\text{uav}} \right\} \tag{26}$$

Static threat regions in the environment are modeled as circular no-fly zones in the horizontal projection plane. Suppose that the environment contains K obstacles, where the k-th obstacle is defined by its center location $O_k(x_k, y_k)$ and influence radius R_k. The geometric parameters and spatial locations of all obstacles are summarized in Table 14. During collision avoidance and safety distance evaluation, only geometric relationships in the horizontal plane are considered. For a trajectory point $P(x, y)$, the condition for satisfying the safety distance requirement is defined in Eq. 26, where R_{uav} denotes the effective horizontal radius of the UAV. This condition forms the geometric basis for the threat avoidance term in the subsequent cost function formulation.

Table 14: Geometric parameters of static obstacles.

Index	Center Coordinate (geospatial)	Radius
Obstacle1	[400, 500, 100]	80
Obstacle2	[600, 200, 150]	70
Obstacle3	[500, 350, 150]	80
Obstacle4	[350, 200, 150]	70
Obstacle5	[700, 550, 150]	70
Obstacle6	[650, 750, 150]	80

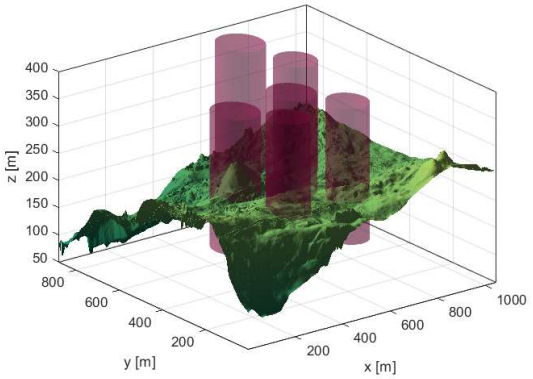


Fig. 11: Spatial distribution of static obstacles in the experimental environment.

5.2 Incremental Trajectory Encoding Based on Spherical Coordinates

To reduce the dimensionality of decision variables and preserve geometric continuity in three-dimensional trajectory planning, an incremental trajectory encoding scheme

based on spherical coordinates is adopted. Instead of directly optimizing absolute spatial coordinates, a trajectory is represented as a sequence of N consecutive path segments, each parameterized by a step length r_i, a pitch angle ψ_i, and a yaw angle ϕ_i. The position of each trajectory node is computed recursively from its predecessor, thereby ensuring spatial continuity along the flight path. Specifically, the three-dimensional coordinates of the i-th trajectory node are updated according to Eq. 27, where x_{i-1} y_{i-1} z_{i-1} denotes the position of the preceding node.

$$\begin{cases} x_i = x_{i-1} + r_i \cos\psi_i \sin\phi_i, \\ y_i = y_{i-1} + r_i \cos\psi_i \cos\phi_i, \\ z_i = z_{i-1} + r_i \sin\psi_i. \end{cases} \tag{27}$$

During optimization, bounds are imposed on the decision variables to constrain the search space and maintain flight feasibility. The step length is restricted to $r_i \in [0, r_{\max}]$, limiting the maximum spatial displacement between consecutive nodes. The pitch angle satisfies $\psi_i \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$, thereby constraining ascent and descent rates. The yaw angle is limited to $\phi_i \in \left[\phi_{\text{goal}} - \frac{\pi}{4}, \phi_{\text{goal}} + \frac{\pi}{4}\right]$,. where ϕ_{goal} denotes the reference direction from the start point toward the target. This constraint restricts horizontal search directions to a goal-oriented angular sector, preventing excessive exploration away from the destination.

5.3 Objective Function and Constraint Handling

Based on the established environment modeling and incremental trajectory parameterization, the three-dimensional UAV trajectory planning problem is formulated as a single-objective minimization problem with multiple weighted components. The overall cost function J consists of four sub-cost terms, namely path length, obstacle threat, flight altitude regulation, and trajectory smoothness. Their weighted combination is defined in Eq. 28. where $b_1 \sim b_4$ denote the weighting coefficients used to regulate the relative contribution of each cost component.

$$J = b_1 J_1 + b_2 J_2 + b_3 J_3 + b_4 J_4 \tag{28}$$

The mathematical formulations of the individual cost components are described as follows:

Path length cost J_1: The path length cost characterizes the total travel distance of the UAV trajectory and is computed by accumulating the Euclidean distances between consecutive trajectory nodes in three-dimensional space. As defined in Eq. 29, the absolute flight altitude Z_{abs}, which incorporates terrain elevation, is used to ensure that terrain undulations are reflected in the distance calculation.

$$J_1 = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + \left(Z_{\text{abs}}^{(i+1)} - Z_{\text{abs}}^{(i)}\right)^2} \tag{29}$$

Obstacle threat cost J_2: To avoid potential path penetration caused by collision checking only at discrete trajectory nodes, a segment-based minimum distance evaluation strategy is adopted. Specifically, for each trajectory segment connecting

P_i and P_{i+1}, the minimum horizontal distance d_{ik} between the segment and the center of the k-th obstacle O_k is computed in the horizontal projection plane. Based on this distance, a piecewise obstacle penalty function $\mathrm{Cost}_{\mathrm{obs}}(d_{ik})$ is defined, as shown in Eq. 30. When $d_{ik} < R_k + R_{\mathrm{uav}}$, the trajectory segment is regarded as colliding with the obstacle and is assigned a large infeasible penalty J_{inf}. When $R_k + R_{\mathrm{uav}} \leq d_{ik} \leq R_k + R_{\mathrm{uav}} + D_{\mathrm{sense}}$, a linearly increasing penalty is imposed to form a soft safety buffer. If $d_{ik} > R_k + R_{\mathrm{uav}} + D_{\mathrm{sense}}$, no penalty is applied. The total obstacle threat cost is then obtained by summing the penalties over all segments and obstacles, as defined in Eq. 31.

$$\mathrm{Cost}_{\mathrm{obs}}(d) = \begin{cases} J_{\mathrm{inf}}, & d < R_k + R_{\mathrm{uav}}, \\ (R_k + R_{\mathrm{uav}} + D_{\mathrm{sense}}) - d, & R_k + R_{\mathrm{uav}} \leq d \leq R_k + R_{\mathrm{uav}} + D_{\mathrm{sense}}, \\ 0, & d > R_k + R_{\mathrm{uav}} + D_{\mathrm{sense}}. \end{cases} \tag{30}$$

$$J_2 = \sum_{k=1}^{K} \sum_{i=1}^{N-1} \mathrm{Cost}_{\mathrm{obs}}(d_{ik}) \tag{31}$$

Flight altitude cost J_3: The flight altitude cost encourages the UAV to maintain its relative altitude within a desired interval $z_{\min} z_{\max}$. It is defined in Eq. 32, where z_i denotes the relative altitude at the i-th trajectory node. When $z_i \leq 0$, a ground collision is assumed and a large infeasible penalty J_{inf} is assigned to eliminate the corresponding trajectory.

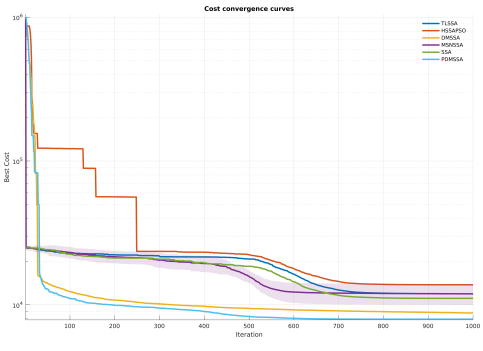
$$J_3 = \sum_{i=1}^{N} \left| z_i - \frac{z_{\max} + z_{\min}}{2} \right| \tag{32}$$

Trajectory smoothness cost J_4: To suppress abrupt directional changes between adjacent trajectory segments, a smoothness cost based on angular variations is introduced. As defined in, Eq. 33 this term penalizes the horizontal turning angle α_i and the change in climb angle $\Delta\beta_i$ between consecutive segments. The indicator function $I(\cdot)$ contributes to the cost only when the corresponding angular variation exceeds predefined thresholds, thereby promoting trajectory smoothness without over-constraining gradual directional changes.

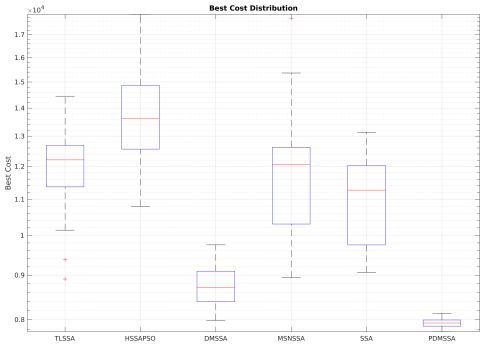
$$J_4 = \sum_{i=1}^{N-2} \left(|\alpha_i| \cdot I(|\alpha_i| > \alpha_{\max}) + |\Delta\beta_i| \cdot I(|\Delta\beta_i| > \beta_{\max}) \right) \tag{33}$$

5.4 Results Analysis

To evaluate the performance of the proposed PDMSSA on the three-dimensional UAV trajectory planning problem, 30 independent runs are conducted and the results are compared with TLSSA, HSSAPSO, DMSSA, MSNSSA, and the standard SSA. The



(a) Convergence curves of the optimal flight cost.



(b) Boxplot of the optimal flight cost.

Fig. 12: Result analysis of the optimal flight cost.

comparison is performed in terms of statistical performance indicators and convergence behavior. Table 15 reports the minimum, maximum, and mean flight costs obtained by each algorithm over 30 independent runs. Under the same environment and parameter settings, PDMSSA achieves the lowest mean cost and minimum cost among all compared algorithms, while its maximum cost remains at a relatively low level. These results indicate that PDMSSA provides competitive solution quality with reduced performance variability across independent executions. The average convergence curves shown in Fig. 12a illustrate the cost reduction process of different algorithms. PDMSSA exhibits a steady and sustained decrease in the objective value throughout the iterations and converges to a lower final cost compared with the competing methods. In contrast, several comparison algorithms show slower convergence or early stagnation at higher cost levels. The boxplots in Fig. 12b further depict the distribution of final costs over 30 runs. PDMSSA presents a lower median cost with a relatively compact interquartile range and no evident outliers, indicating stable optimization behavior and reduced sensitivity to random initialization. Fig. 13 visualizes representative three-dimensional flight paths generated by different algorithms in the constructed environment. The trajectory obtained by PDMSSA maintains feasible clearance from obstacles and follows a smooth path over the terrain, which is consistent with the statistical results.

Overall, the experimental results demonstrate that, within the unified environment modeling and cost formulation framework, PDMSSA is able to stably generate feasible UAV trajectories with lower comprehensive cost and improved consistency. These findings suggest that the proposed method is effective for complex three-dimensional UAV trajectory planning problems involving terrain variation and obstacle constraints.

6 Conclusions and Future Work

To address the inherent limitations of the Salp Swarm Algorithm (SSA), such as excessive reliance on dominant leaders, insufficient directional guidance, and search

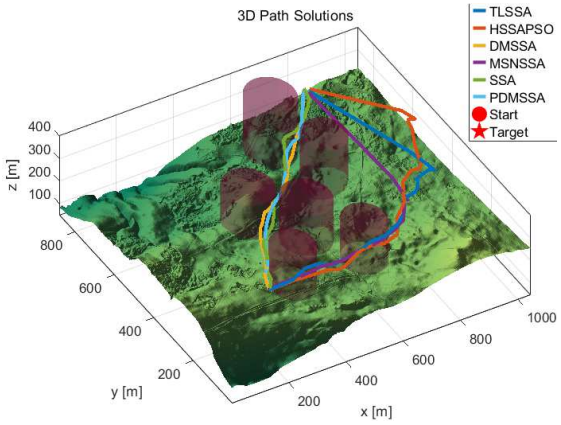


Fig. 13: Optimal Flight Path Map.

Table 15: Statistical cost comparison of different algorithms.

Row	TLSSA	HSSAPSO	DMSSA	MSNSSA	SSA	PDMSSA
MinCost	8.58×10^3	1.12×10^4	8.04×10^3	8.75×10^3	8.56×10^3	7.75×10^3
MaxCost	1.39×10^4	1.79×10^4	9.72×10^3	1.78×10^4	2.02×10^4	8.13×10^3
MeanCost	1.16×10^4	1.38×10^4	8.69×10^3	1.23×10^4	1.11×10^4	7.94×10^3

stagnation in later evolutionary stages, this paper three core refinement strategies to boost the performance of SSA, and the refined algorithm named PDMSSA. The first strategy is the Pareto-guided leader selection strategy, which conducts fast non-dominated sorting based on both the solutions' fitness values and their spatial distribution to ensure leaders are both high-quality and widely distributed in the search space. The scale-free network-based multi-source information fusion mechanism is the second core strategy to refine position updating of followers with the joint guidance of their network-selected neighbor and elite reference solutions. At last, a state-aware directional tabu search strategy is proposed to suppress redundant search behaviors and facilitating targeted escape from stagnation.

A convergence analysis conducted on the CEC2017 and CEC2022 benchmark test suites demonstrate that PDMSSA achieves competitive and stable performance across different function categories and dimensional settings when compared with several representative metaheuristic algorithms and SSA variants. In particular, PDMSSA maintains relatively robust optimization performance in high-dimensional multimodal problems. Further validation on constrained engineering design problems and three-dimensional UAV trajectory planning tasks indicates that the proposed algorithm can stably generate feasible solutions under complex constraints, highlighting its practical applicability. Despite these advantages, PDMSSA introduces additional computational overhead due to Pareto-based sorting and network-related operations, which may become non-negligible for large population sizes. Moreover, some parameters in the directional tabu search mechanism rely on empirical configuration. Future work

will focus on reducing computational overhead through more efficient implementations, developing adaptive parameter control strategies, and extending the proposed framework to discrete and multi-objective optimization problems.

Statements and Declarations

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61563019 and 61562037), and the Jiangxi University of Science and Technology funded project (205200100013).

Funding

National Natural Science Foundation of China (61563019 and 61562037), Jiangxi University of Science and Technology funded project (205200100013).

Declaration of competing interest

The authors declare that there are no competing interests regarding the publication of this paper.

CRediT authorship contribution statement

Dahai Li: Conceptualization, Data curation, Writing – review & editing.
Zexiong Liu: Methodology, Software, Investigation, Writing – original draft.
Zhendong Wang: Visualization, Supervision.
All authors have read and approved the final manuscript.

Data availability

The benchmark test suites (CEC2017 and CEC2022) utilized in this study are standardized, publicly available datasets. The implementation code of PDMSSA is available at https://github.com/lpxks/PDMSSA. No additional datasets were generated or analyzed during the current study.

References

[1] Meza, J.C.: Newton's method. Wiley Interdisciplinary Reviews: Computational Statistics **3**(1), 75–78 (2011) https://doi.org/10.1002/wics.129

[2] Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016) https://doi.org/10.48550/arXiv.1609.04747

[3] Rajwar, K., Deep, K., Das, S.: An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artificial Intelligence Review **56**(11), 13187–13257 (2023) https://doi.org/10.1007/s10462-023-10470-y

[4] Li, G., Zhang, T., Tsai, C.-Y., Yao, L., Lu, Y., Tang, J.: Review of the metaheuristic algorithms in applications: Visual analysis based on bibliometrics. Expert Systems with Applications **255**, 124857 (2024) https://doi.org/10.1016/j.eswa.2024.124857

[5] Slowik, A., Kwasnicka, H.: Evolutionary algorithms and their applications to engineering problems. Neural Computing and Applications **32**(16), 12363–12379 (2020) https://doi.org/10.1007/s00521-020-04832-8

[6] Mathew, T.V.: Genetic algorithm. Report submitted at IIT Bombay **53**, 18–19 (2012) https://doi.org/10.1145/234313.23435

[7] Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization **11**(4), 341–359 (1997) https://doi.org/10.1023/a:1008202821328

[8] Seyyedabbasi, A.: A comprehensive survey: Physics-based algorithms. Decision-Making Models, 59–75 (2024) https://doi.org/10.1016/B978-0-443-16147-6.00033-5

[9] Bertsimas, D., Tsitsiklis, J.: Simulated annealing. Statistical science **8**(1), 10–15 (1993) https://doi.org/10.1214/ss/1177011077

[10] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. Information sciences **179**(13), 2232–2248 (2009) https://doi.org/10.1016/j.ins.2009.03.004

[11] Patel, V.K., Savsani, V.J.: Heat transfer search (hts): a novel optimization algorithm. Information sciences **324**, 217–246 (2015) https://doi.org/10.1016/j.ins.2015.06.044

[12] Liu, Y., Li, R.: Psa: a photon search algorithm. Journal of Information Processing Systems **16**(2), 478–493 (2020) https://doi.org/10.3745/JIPS.04.0168

[13] Hatamlou, A.: Black hole: A new heuristic optimization approach for data clustering. Information sciences **222**, 175–184 (2013) https://doi.org/10.1016/j.ins.2012.08.023

[14] Ghafari, R., Mansouri, N.: Swarm intelligence techniques and their applications in fog/edge computing: an in-depth review. Artificial Intelligence Review **58**(11), 359 (2025) https://doi.org/10.1007/s10462-025-11351-2

[15] Cetinsaya, B., Reiners, D., Cruz-Neira, C.: From pid to swarms: A decade of advancements in drone control and path planning-a systematic review (2013–2023). Swarm and Evolutionary Computation **89**, 101626 (2024) https://doi.org/10.1016/j.swevo.2024.101626

[16] Yang, X.-S., Deb, S., Zhao, Y.-X., Fong, S., He, X.: Swarm intelligence: past, present and future. Soft Computing **22**(18), 5923–5933 (2018) https://doi.org/10.1007/s00500-017-2810-5

[17] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-international Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995). https://doi.org/10.1109/ICNN.1995.488968 . ieee

[18] Jia, H., Zhou, X., Zhang, J., Mirjalili, S.: Superb fairy-wren optimization algorithm: a novel metaheuristic algorithm for solving feature selection problems. Cluster Computing **28**(4), 246 (2025) https://doi.org/10.1007/s10586-024-04901-w

[19] Dehghani, M., Montazeri, Z., Trojovská, E., Trojovskỳ, P.: Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Knowledge-based systems **259**, 110011 (2023) https://doi.org/10.1016/j.knosys.2022.110011

[20] Xue, J., Shen, B.: Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. The journal of Supercomputing **79**(7), 7305–7336 (2023) https://doi.org/10.1007/s11227-022-04959-6

[21] El-Kenawy, E.-S.M., Khodadadi, N., Mirjalili, S., Abdelhamid, A.A., Eid, M.M., Ibrahim, A.: Greylag goose optimization: nature-inspired optimization algorithm. Expert Systems with Applications **238**, 122147 (2024) https://doi.org/10.1016/j.eswa.2023.122147

[22] Mirjalili, S., Lewis, A.: The whale optimization algorithm. Advances in engineering software **95**, 51–67 (2016) https://doi.org/10.1016/j.advengsoft.2016.01.008

[23] Guo, Z., Liu, G., Jiang, F.: Chinese pangolin optimizer: A novel bio-inspired metaheuristic for solving optimization problems. The Journal of Supercomputing **81**(4), 517 (2025) https://doi.org/10.1007/s11227-025-07004-4

[24] Xue, J., Shen, B.: A novel swarm intelligence optimization approach: sparrow search algorithm. Systems science & control engineering **8**(1), 22–34 (2020) https://doi.org/10.1080/21642583.2019.1708830

[25] Abdollahzadeh, B., Gharehchopogh, F.S., Mirjalili, S.: African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Computers & Industrial Engineering **158**, 107408 (2021) https://doi.org/10.1016/j.cie.2021.107408

[26] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software **114**, 163–191 (2017) https://doi.org/10.1016/

j.advengsoft.2017.07.002

[27] Zhang, H., Liu, T., Ye, X., Heidari, A.A., Liang, G., Chen, H., Pan, Z.: Differential evolution-assisted salp swarm algorithm with chaotic structure for real-world problems. Engineering with Computers **39**(3), 1735–1769 (2023) https://doi.org/10.1007/s00366-021-01545-x

[28] Tawhid, M.A., Ibrahim, A.M.: Improved salp swarm algorithm combined with chaos. Mathematics and Computers in Simulation **202**, 113–148 (2022) https://doi.org/10.1016/j.matcom.2022.05.029

[29] El-Shorbagy, M.A., Eldesoky, I.M., Basyouni, M.M., Nassar, I., El-Refaey, A.M.: Chaotic search-based salp swarm algorithm for dealing with system of nonlinear equations and power system applications. Mathematics **10**(9), 1368 (2022) https://doi.org/10.3390/math10091368

[30] Wang, Z., Ding, H., Yang, J., Hou, P., Dhiman, G., Wang, J., Yang, Z., Li, A.: Orthogonal pinhole-imaging-based learning salp swarm algorithm with self-adaptive structure for global optimization. Frontiers in Bioengineering and Biotechnology **10**, 1018895 (2022) https://doi.org/10.3389/fbioe.2022.1018895

[31] Fan, Q., Zhao, S., Shang, M., Wei, Z., Huang, X.: An improved genetic salp swarm algorithm with population partitioning for numerical optimization. Information Sciences **679**, 120895 (2024) https://doi.org/10.1016/j.ins.2024.120895

[32] Zhong-Yun, C., Da-Min, Z., Zi-Yun, X.: Multi-subpopulation based symbiosis and non-uniform gaussian mutation salp swarm algorithm. ACTA automatica sinica, 1–9 (2021) https://doi.org/10.16383/j.aas.c190684

[33] Yao, J., Sha, Y., Chen, Y., Zhang, G., Hu, X., Bai, G., Liu, J.: Ihssao: An improved hybrid salp swarm algorithm and aquila optimizer for uav path planning in complex terrain. Applied Sciences **12**(11), 5634 (2022) https://doi.org/10.3390/app12115634

[34] Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A.A., Al-Qaness, M.A., Gandomi, A.H.: Aquila optimizer: a novel meta-heuristic optimization algorithm. Computers & Industrial Engineering **157**, 107250 (2021) https://doi.org/10.1016/j.cie.2021.107250

[35] Singh, N., Singh, S., Houssein, E.H.: Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions. Evolutionary Intelligence **15**(1), 23–56 (2022) https://doi.org/10.1007/s12065-020-00486-6

[36] Mohamed, A.A., Abdellatif, A.D., Alburaikan, A., Khalifa, H.A.E.-W., Elaziz, M.A., Abualigah, L., AbdelMouty, A.M.: A novel hybrid arithmetic optimization algorithm and salp swarm algorithm for data placement in cloud computing. Soft Computing **27**(9), 5769–5780 (2023) https://doi.org/10.1007/

s00500-022-07805-2

[37] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A.H.: The arithmetic optimization algorithm. Computer methods in applied mechanics and engineering **376**, 113609 (2021)

[38] Chauhan, S., Vashishtha, G., Abualigah, L., Kumar, A.: Boosting salp swarm algorithm by opposition-based learning concept and sine cosine algorithm for engineering design problems. Soft Computing **27**(24), 18775–18802 (2023) https://doi.org/10.1007/s00500-023-09147-z

[39] Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. Knowledge-based systems **96**, 120–133 (2016) https://doi.org/10.1016/j.knosys.2015.12.022

[40] Li, J., Ren, H., Chen, H., Li, C.: Teaching–learning guided salp swarm algorithm for global optimization tasks and feature selection. Soft Computing **27**(23), 17887–17908 (2023) https://doi.org/10.1007/s00500-023-09070-3

[41] Lin, C., Wang, P., Zhao, X., Chen, H.: Double mutational salp swarm algorithm: from optimal performance design to analysis. Journal of Bionic Engineering **20**(1), 184–211 (2023) https://doi.org/10.1007/s42235-022-00262-5

[42] Sulaiman, M.H., Mustaffa, Z.: An application of improved salp swarm algorithm for optimal power flow solution considering stochastic solar power generation. e-Prime-Advances in Electrical Engineering, Electronics and Energy **5**, 100195 (2023) https://doi.org/10.1016/j.prime.2023.100195

[43] Abualigah, L., Al-Okbi, N.K., Elaziz, M.A., Houssein, E.H.: Boosting marine predators algorithm by salp swarm algorithm for multilevel thresholding image segmentation. Multimedia tools and applications **81**(12), 16707–16742 (2022) https://doi.org/10.1007/s11042-022-12001-3

[44] Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H.: Marine predators algorithm: A nature-inspired metaheuristic. Expert systems with applications **152**, 113377 (2020) https://doi.org/10.1016/j.eswa.2020.113377

[45] Zhao, S., Wang, P., Heidari, A.A., Chen, H., He, W., Xu, S.: Performance optimization of salp swarm algorithm for multi-threshold image segmentation: Comprehensive study of breast cancer microscopy. Computers in biology and medicine **139**, 105015 (2021) https://doi.org/10.1016/j.compbiomed.2021.105015

[46] Mahajan, S., Mittal, N., Salgotra, R., Masud, M., Alhumyani, H.A., Pandit, A.K.: An efficient adaptive salp swarm algorithm using type ii fuzzy entropy for multilevel thresholding image segmentation. Computational and Mathematical Methods in Medicine **2022**(1), 2794326 (2022) https://doi.org/10.1155/2022/2794326

[47] Dagal, I., Akın, B., Akboy, E.: A novel hybrid series salp particle swarm optimization (sspso) for standalone battery charging applications. Ain Shams Engineering Journal **13**(5), 101747 (2022) https://doi.org/10.1016/j.asej.2022.101747

[48] Rustagi, K., Bhatnagar, P., Mathur, R., Singh, I., G, S.K.: Hybrid salp swarm and grey wolf optimizer algorithm based ensemble approach for breast cancer diagnosis. Multimedia Tools and Applications **83**(27), 70117–70141 (2024) https://doi.org/10.1007/s11042-023-18015-9

[49] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in engineering software **69**, 46–61 (2014) https://doi.org/10.1016/j.advengsoft.2013.12.007

[50] Abualigah, L., Shehab, M., Alshinwan, M., Alabool, H.: Salp swarm algorithm: a comprehensive survey. Neural Computing and Applications **32**(15), 11195–11215 (2020) https://doi.org/10.1007/s00521-019-04629-4

[51] Aljarah, I., Mafarja, M., Heidari, A.A., Faris, H., Zhang, Y., Mirjalili, S.: Asynchronous accelerating multi-leader salp chains for feature selection. Applied Soft Computing **71**, 964–979 (2018) https://doi.org/10.1016/j.asoc.2018.07.040

[52] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002) https://doi.org/10.1109/4235.996017

[53] Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. science **286**(5439), 509–512 (1999) https://doi.org/10.1126/science.286.5439.509

[54] Glover, F.: Tabu search—part i. ORSA Journal on computing **1**(3), 190–206 (1989) https://doi.org/10.1287/ijoc.1.3.190

[55] Glover, F.: Tabu search—part ii. ORSA Journal on computing **2**(1), 4–32 (1990) https://doi.org/10.1287/ijoc.2.1.4

[56] Salgotra, R., Sharma, P., Kundu, K., Raju, S., Gandomi, A.H.: Enhancing differential evolution algorithm for cec 2014, cec 2017, cec 2021, and cec 2022 test suites. Neural Computing and Applications **37**(33), 27593–27630 (2025) https://doi.org/10.1007/s00521-025-11678-5

[57] Biedrzycki, R.: Analysis and simplification of the winner of the cec 2022 optimization competition on single objective bound constrained search. Evolutionary Computation, 1–19 (2025) https://doi.org/10.1162/evco.a.27