symmetry

MDPI

Article

Short-Term Power Load Forecasting Model Based on an Adaptive Multi-Strategy Gold Rush Optimizer for Optimizing an Dilated BiGRU

Xiyuan Li [1] and Yangjian Yang [2,*]

[1] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore; lxy94673197@gmail.com
[2] Polytechnic Institute, Zhejiang University, Hangzhou 310015, China
* Correspondence: yj_yang@zju.edu.cn

Abstract

Short-term power load forecasting plays a crucial role in the operation scheduling, energy management, and security assessment of smart grids, as its accuracy directly affects the economic efficiency and reliability of power systems. However, power load series are characterized by strong nonlinearity, non-stationarity, and multi-scale temporal dependencies, which make it difficult for traditional forecasting models to achieve both high accuracy and strong generalization under complex load scenarios. In recent years, deep learning models have demonstrated promising performance in load forecasting; nevertheless, their effectiveness is highly dependent on hyperparameter configurations. Manual hyperparameter tuning is not only time-consuming but also prone to premature convergence to suboptimal solutions. To overcome these challenges, this study develops an AMS-GRO-based short-term power load forecasting framework. By improving the original Gold Rush Optimizer (GRO), three adaptive strategies are integrated into the proposed optimizer, namely the current-to-pbest/1 mutation strategy, adaptive elite-guided search strategy, and success-rate-based adaptive strategy selection mechanism. These improvements jointly enhance the global exploration capability, local exploitation ability, and strategy adaptability of the optimizer. Subsequently, the proposed AMS-GRO is employed to perform global hyperparameter optimization for a Dilated BiGRU-Attention model, forming the AMS-GRO–Dilated BiGRU forecasting framework. Key hyperparameters, including the learning rate, number of hidden units, attention dimension, and regularization coefficient, are adaptively optimized. Extensive numerical experiments were conducted on the 30-dimensional CEC2017 suite and the 10- and 20-dimensional CEC2022 suites using 30 independent runs. According to the Friedman test, AMS-GRO achieved mean ranks of 1.37, 1.33, and 1.25, respectively, ranking first in all three experimental settings. In the short-term power load forecasting experiment, AMS-GRO–Dilated BiGRU achieved an MAE of 21.09, MAPE of 0.0177, MSE of 791.10, and R^2 of 0.9746. Compared with the unoptimized Dilated BiGRU model, these results correspond to reductions of 13.0%, 13.7%, and 21.5% in MAE, MAPE, and MSE, respectively, together with a 0.70-percentage-point improvement in R^2. From a symmetry perspective, the proposed framework balances global exploration and local exploitation through adaptive strategy selection, while the Dilated BiGRU exploits bidirectional temporal symmetry to capture multi-scale load patterns. These results demonstrate that the proposed framework provides competitive optimization performance and improves forecasting accuracy on the investigated load dataset.

Check for updates

Academic Editor: Yirui Wang

Received: 4 September 2026
Revised: 16 September 2026
Accepted: 22 September 2026
Published: 29 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: short-term power load forecasting; adaptive multi-strategy gold rush optimizer; Dilated BiGRU; hyperparameter optimization; global optimization

1. Introduction

As a critical enabling technology underpinning operational dispatch, grid infrastructure planning, and energy resource management within power systems, short-term load forecasting (STLF) directly influences the security, cost-effectiveness, and overall energy efficiency of grid operations [1]. The ongoing large-scale integration of variable renewable energy sources, together with the growing diversity of consumption patterns among end-users and the rising incidence of severe weather events, has rendered the short-term load—conventionally projected over a horizon spanning from 1 to 72 h—increasingly volatile, inherently nonlinear, and governed by intricate multi-scale temporal interdependencies [2,3]. Prolonged heatwave episodes, for instance, tend to precipitate a surge in cooling demand from air-conditioning systems, while the inherent variability and unpredictability of wind and solar generation further complicate the real-time balancing between generation output and load demand. These compounding factors cause the underlying statistical properties of load time series to shift continuously over time, thereby adding substantial complexity to the construction of accurate predictive models.

Traditional forecasting approaches based on time-series analysis or linear regression models struggle to effectively capture these complex characteristics and often produce large prediction errors under highly volatile operating conditions. Consequently, such methods can no longer meet the practical requirements of modern power systems for refined and intelligent scheduling [4–6].

Over the past several years, deep learning architectures have garnered considerable research interest within the short-term load forecasting community, owing principally to their inherent strengths in capturing complex nonlinear relationships and autonomously distilling high-level features from raw input data. Among these approaches, recurrent neural networks (RNNs), along with their advanced augmented variants—most notably Long Short-Term Memory (LSTM) networks [7] and Gated Recurrent Units (GRUs) [4]—incorporate specialized gating schemes that effectively circumvent the well-known issue of diminishing gradients during backpropagation. These gated architectures have consequently demonstrated substantial promise in uncovering and tracking the sequential regularities embedded in historical load measurements. However, standard LSTM and GRU architectures rely on step-by-step recursive updates, resulting in a receptive field that grows linearly with the time horizon. When dealing with load sequences characterized by long-term periodicity and multi-scale temporal patterns, these models still suffer from limited efficiency in capturing long-range dependencies. Moreover, when the input sequences are long or the load patterns are highly complex, the training process may become slow or even experience performance degradation.

To further enhance the ability of recurrent models to capture long-term temporal dependencies, dilation mechanisms have been introduced into recurrent neural networks, leading to the development of Dilated RNN architectures. Among them, the Dilated Bidirectional Gated Recurrent Unit (Dilated BiGRU) combines bidirectional temporal modeling with dilated connections, enabling an exponential expansion of the receptive field without significantly increasing network depth or parameter complexity. As a result, Dilated BiGRU can simultaneously capture short-term load fluctuations and long-term periodic variations. Existing studies have demonstrated that Dilated BiGRU achieves higher prediction accuracy and better stability than conventional LSTM and GRU models in short-term power

load forecasting tasks [8]. Nevertheless, the prediction performance of Dilated BiGRU is strongly influenced by the selection of hyperparameters, such as the learning rate, hidden unit number, attention dimension, and regularization coefficient. These hyperparameters are often nonlinearly coupled, and relying on manual experience or trial-and-error tuning is not only inefficient but also prone to yielding suboptimal solutions, thereby limiting the generalization ability of the model under highly volatile load scenarios.

Swarm intelligence optimization algorithms, as an important branch of evolutionary computation, simulate cooperative behaviors observed in natural systems to perform global optimization and have demonstrated strong robustness and adaptability in high-dimensional, non-convex, and multimodal search spaces [9–11]. Representative swarm intelligence algorithms include Particle Swarm Optimization (PSO) [12], Grey Wolf Optimization (GWO) [13], Whale Optimization Algorithm (WOA) [14], Secretary Bird Optimization Algorithm (SBOA) [15], Bounty Hunter Optimizer (BHO) [16], Sparrow Search Algorithm (SSA) [17], and Animated Oat Optimization (AOO) [18]. These algorithms are characterized by simple parameter settings, independence from gradient information, and strong compatibility with machine learning models, making them widely applicable to complex optimization problems such as path planning, engineering design, parameter estimation, and feature selection.

Recently, the application of swarm intelligence algorithms to hyperparameter optimization in machine learning and deep learning has become an active research topic. Owing to the high dimensionality and strong nonlinearity of hyperparameter search spaces in deep models, traditional grid search or random search methods often incur high computational costs and low search efficiency. In contrast, swarm intelligence algorithms can effectively explore complex parameter spaces through population-based cooperative search mechanisms, thereby reducing the uncertainty associated with manual tuning [19]. Numerous studies have shown that integrating swarm intelligence algorithms with deep learning models can significantly improve predictive performance and training stability. For example, PSO and GWO have been successfully applied to hyperparameter optimization of LSTM and GRU models in tasks such as load forecasting, traffic flow prediction, and hydrological forecasting [20,21]. Nevertheless, when applied to high-dimensional hyperparameter tuning tasks, swarm intelligence algorithms that rely on a single optimization strategy continue to encounter persistent difficulties, including a tendency toward premature stagnation at suboptimal solutions, erratic convergence trajectories, and inadequate genetic variety within the candidate solution pool.

The Gold Rush Optimizer (GRO) constitutes a swarm-intelligence-based metaheuristic that has emerged in the recent literature, drawing its conceptual foundations from the migratory patterns, extraction activities, and cooperative interactions observed among historical gold prospectors [22]. GRO is characterized by a parsimonious algorithmic architecture, requiring only a limited set of configurable parameters and offering straightforward deployment, which has contributed to its competitive empirical performance across a broad spectrum of engineering-oriented optimization tasks. However, when confronting intricate high-dimensional search spaces, the algorithm reveals several nontrivial shortcomings. Chief among these is the absence of a well-designed adaptive regulation mechanism capable of dynamically adjusting the trade-off between broad global exploration and intensive local exploitation as the search process unfolds. This inherent deficiency frequently manifests as oscillatory patterns in the convergence trajectory or, alternatively, as a premature settling toward locally optimal regions during the later stages of iteration. Second, the individual update strategies are relatively simple, resulting in limited utilization efficiency of high-quality solutions and difficulty in maintaining both population diversity and convergence speed in complex search spaces [23,24]. In addition, fixed or weakly adaptive strategy

selection schemes prevent the algorithm from dynamically allocating search resources based on real-time feedback from the optimization process, thereby restricting its overall optimization performance.

To overcome these limitations, this paper proposes an Adaptive Multi-Strategy Gold Rush Optimizer (AMS-GRO) and integrates it into the global hyperparameter optimization of a Dilated BiGRU model, forming an AMS-GRO–Dilated BiGRU short-term power load forecasting framework. By incorporating multiple cooperative search strategies and a dynamic selection mechanism. The proposed AMS-GRO variant delivers substantial improvements across three key performance dimensions: the breadth of global search, the precision of local refinement, and the robustness of the convergence process. The principal innovations and contributions of this work are articulated below:

(1) Current-to-pbest/1–based convergence–diversity balanced mutation strategy: A current-to-pbest/1 mutation mechanism is introduced into the GRO framework to accelerate convergence while effectively preserving population diversity, thereby alleviating premature convergence.
(2) Adaptive elite-guided search strategy: By restricting learning sources to an elite subset and introducing adaptive perturbation control, the proposed strategy enhances local exploitation capability and improves convergence stability in the later stages of optimization.
(3) Success-rate–based adaptive strategy selection mechanism: The selection probability of each search strategy is dynamically adjusted according to its actual contribution during the optimization process, enabling adaptive cooperation and competition among multiple strategies.
(4) Global hyperparameter optimization framework for Dilated BiGRU: AMS-GRO is employed to optimize key hyperparameters of the Dilated BiGRU model, constructing an integrated "intelligent optimization–deep temporal modeling" forecasting framework that eliminates the reliance on manual hyperparameter tuning.

To evaluate the performance of the proposed method, extensive numerical experiments are first conducted on the CEC2022 and CEC2017 benchmark suites. Subsequently, the Friedman test is performed to statistically analyze the differences among the compared algorithms. Following this, forecasting experiments are carried out on short-term power load using authentic electricity consumption records from real-world grid operations. The predictive capability of the AMS-GRO–Dilated BiGRU hybrid is benchmarked against several alternative architectures, including RNN, LSTM, KELM, and the standalone Dilated BiGRU without optimization, with its performance being systematically quantified through the MAE, MAPE, MSE, and R^2 evaluation criteria.

The concept of symmetry is also relevant to the design of the proposed method. In metaheuristic optimization, exploration and exploitation are two complementary search behaviors that must be balanced dynamically; this balance can be regarded as a form of behavioral symmetry in the search process. The proposed AMS-GRO maintains such a symmetry by adaptively allocating search resources among multiple strategies according to their real-time success rates. In addition, the Dilated BiGRU processes load sequences in both forward and backward directions, which introduces a bidirectional temporal symmetry and allows the model to exploit past and future contextual information simultaneously. The dilation mechanism further enables multi-scale temporal features to be cap-tured in a balanced and hierarchical manner. Therefore, symmetry, in both its behavioral and temporal senses, provides a useful perspective for understanding the design rationale of the proposed forecasting framework.

The structural layout of the present paper proceeds in the following sequence. Section 2 elaborates on the foundational underpinnings of both the canonical Gold

Rush Optimizer and the newly devised AMS-GRO variant. Section 3 details a battery of numerical tests designed to scrutinize the optimization proficiency of the enhanced algorithm. Section 4 formulates the integrated forecasting framework that combines AMS-GRO with Dilated BiGRU for short-term load prediction, accompanied by a thorough examination of the experimental findings. Section 5, in closing, synthesizes the principal conclusions drawn from this investigation and identifies promising avenues for subsequent research endeavors.

2. The Gold Rush Optimizer and Its Enhanced Formulation

2.1. Canonical Gold Rush Optimizer

The Gold Rush Optimizer (GRO) belongs to the category of population-driven metaheuristic search techniques, with its underlying design philosophy rooted in the collective behavioral patterns exhibited by historical gold seekers throughout their extraction endeavors. Within the GRO framework, every individual prospector corresponds to a potential solution within the decision space, while the overall optimization procedure is orchestrated through the coordinated execution of three distinct strategies—namely, migratory movement, localized excavation, and inter-prospector cooperation [22].

Let $X_i^{(t)} = \left[x_{i,1}^{(t)}, x_{i,2}^{(t)}, \ldots, x_{i,D}^{(t)}\right]$ represent the position of the i-th prospector at iteration t. Here, D denotes the dimensionality of the optimization problem, and N represents the population size. The historically optimal positional vector identified up to the current iteration t is designated as $X_{best}^{(t)}$.

(1) Migration behavior

The migration behavior simulates the movement of prospectors toward a discovered gold mine. In this stage, each prospector updates its position by moving toward the current global best solution with a randomly scaled step size. The position update can be expressed as [22]:

$$X_i^{(t+1)} = X_i^{(t)} + A_1^{(t)} \cdot \left(C_1^{(t)} \cdot X_{\text{best}}^{(t)} - X_i^{(t)}\right) \tag{1}$$

where $A_1^{(t)}$ and $C_1^{(t)}$ are stochastic control coefficients.

Specifically, $C_1^{(t)}$ controls the attraction strength toward the global best solution and is generated by a uniformly distributed random variable in (0, 2), while $A_1^{(t)}$ determines the step size and is adaptively adjusted according to the iteration progress using a nonlinear decreasing function combined with a random perturbation. This design enables strong global exploration in early iterations and more stable convergence behavior in later stages.

(2) Mining behavior

After prospectors reach a potentially gold-rich region, local mining is performed to exploit the surrounding area. In this behavior, each prospector updates its position relative to a randomly selected prospector [22]:

$$X_i^{(t+1)} = X_r^{(t)} + A_2^{(t)} \cdot \left(X_i^{(t)} - X_r^{(t)}\right) \tag{2}$$

where $r \neq i$ is a randomly selected index, and $A_2^{(t)}$ is a stochastic coefficient that controls the exploitation intensity.

The coefficient $A_2^{(t)}$ is generated using a nonlinear decreasing function similar to $A_1^{(t)}$, allowing the algorithm to gradually shift from coarse local exploration to fine-grained exploitation as the iteration proceeds.

(3) Collaboration behavior

The collaboration behavior models cooperative exploration among prospectors. Two randomly selected prospectors are used to construct a cooperative search direction, and the position of the i-th prospector is updated as [22]:

$$X_i^{(t+1)} = X_i^{(t)} + r \cdot \left(X_{g_2}^{(t)} - X_{g_1}^{(t)} \right) \tag{3}$$

where g_1 and g_2 are two distinct prospectors selected at random, and $r \sim U(0,1)$ is a uniformly distributed random number.

This mechanism introduces additional population diversity and helps the algorithm escape from local optima by encouraging cooperative exploration.

(4) Selection mechanism

After generating a candidate solution using one of the above behaviors, GRO adopts a greedy selection strategy to determine whether the new position should be accepted:

$$X_i = \begin{cases} X_i^{(t+1)}, & if\ f\left(X_i^{(t+1)}\right) < f\left(X_i^{(t)}\right), \\ X_i^{(t)}, & otherwise. \end{cases} \tag{4}$$

This selection rule ensures that the fitness value of each prospector does not deteriorate over iterations.

2.2. Adaptive Multi-Strategy Gold Rush Optimizer

In response to the deficiencies exhibited by the canonical GRO with respect to convergence robustness and environmental adaptability, an enhanced variant termed the Adaptive Multi-Strategy Gold Rush Optimizer (AMS-GRO) is introduced in this work. The central concept underlying AMS-GRO lies in the establishment of a hybrid search architecture that integrates a suite of complementary exploration and exploitation strategies, which are orchestrated through a unified self-adaptive regulation mechanism that enables both cooperative interaction and competitive selection among the constituent strategies.

Specifically, AMS-GRO integrates three core components:

(1) a convergence–diversity balanced mutation strategy based on current-to-pbest/1;
(2) an adaptive elite-guided search strategy;
(3) a success-rate–based adaptive strategy selection mechanism.

These modules are devised to collectively promote the breadth of global search, the depth of local refinement, and the overall responsiveness of strategy selection as the optimization routine progresses.

2.2.1. Current-to-pbest/1–Based Convergence–Diversity Balanced Mutation Strategy

Position updates in standard GRO are primarily guided by randomly chosen population members or the current global-best solution. Although this mechanism promotes exploration, it offers limited directional guidance toward promising regions. Consequently, the search may converge slowly or fluctuate considerably on complex, high-dimensional problems. AMS-GRO addresses this limitation by introducing a differential-evolution-inspired current-to-pbest/1 mutation strategy. This strategy applies controlled convergence pressure while maintaining population diversity.

At each algorithmic cycle t, the entire swarm is initially ranked in ascending order based on their respective fitness evaluations, after which an elite representative, denoted as

$X_{pbest}^{(t)}$, is stochastically drawn from the uppermost p performers within the ranked list. The cardinality of this elite subset is determined in an adaptive fashion, as specified by

$$p = max\left(2, \left\lfloor N \cdot \left(p_{min} + (p_{max} - p_{min})\left(1 - \frac{t}{T}\right)^{1.2}\right)\right\rfloor\right) \tag{5}$$

where $p_{\min} = 0.05, p_{\max} = 0.20$, N represents the population size, and T indicates the maximum number of iterations.

Based on the selected elite individual, the mutation vector for the i-th prospector is constructed as [25]:

$$\begin{aligned} X_i^{(t+1)} &= X_i^{(t)} + F^{(t)} \cdot \left(X_{pbest}^{(t)} - X_i^{(t)}\right) + F \cdot \left(X_{r_1}^{(t)} - X_{r_2}^{(t)}\right) \\ F^{(t)} &= MF + 0.1 \cdot \tan\left[\pi\left(rand - \frac{1}{2}\right)\right] \end{aligned} \tag{6}$$

where $X_{r_1}^{(t)}$ and $X_{r_2}^{(t)}$ represent two randomly selected and mutually exclusive individuals, while $X_{r_2}^{(t)}$ is selected from the historical archive to maintain population diversity. The scaling factor $F^{(t)}$ is adaptively generated through the parameter control mechanism. MF is selected from a memory pool and updated according to the success history of previous mutations. To ensure numerical stability, $F^{(t)}$ is restricted to the interval [0.05, 1].

By introducing the current-to-pbest/1 strategy, AMS-GRO effectively balances convergence acceleration and diversity preservation, enabling a smooth transition from global exploration to local exploitation.

As show in Figure 1, the term $\left(X_{pbest}^{(t)} - X_i^{(t)}\right)$ introduces explicit convergence pressure toward promising regions, while the differential vector $\left(X_{r_1}^{(t)} - X_{r_2}^{(t)}\right)$ maintains population diversity. The mutation magnitude is adaptively modulated by the scaling coefficient $F^{(t)}$, which serves to facilitate a gradual progression from broad-range global scanning toward fine-grained local refinement over the course of the optimization run.

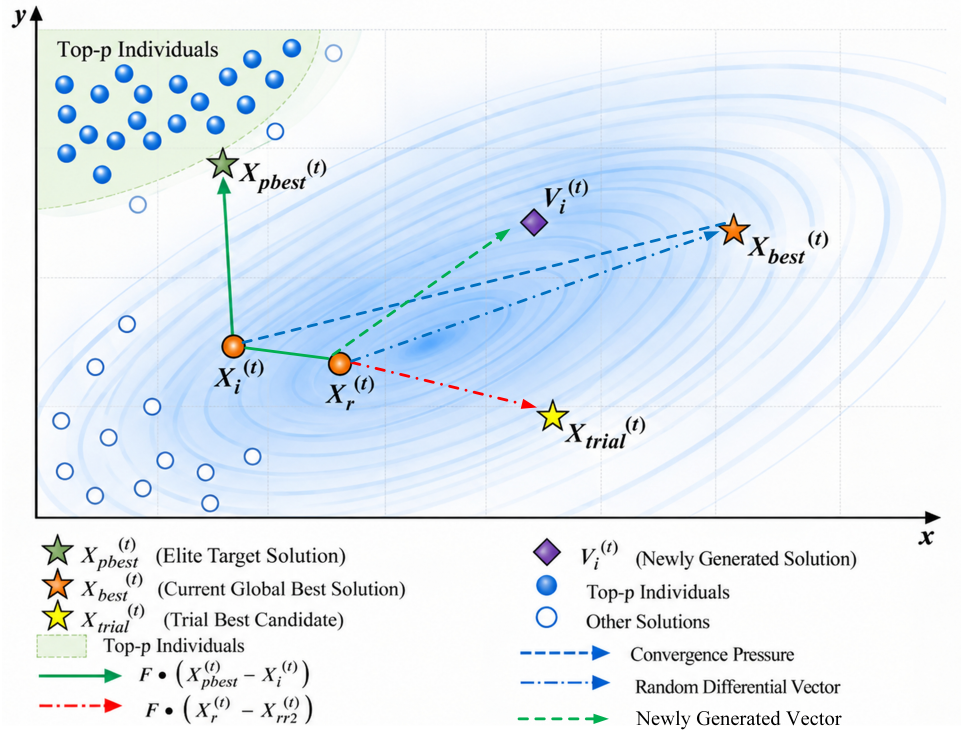


Figure 1. Illustrates the geometric interpretation of the current-to-pbest/1 mutation strategy.

2.2.2. Adaptive Elite-Guided Search (AEG)

Although the stochastic behaviors in GRO provide strong exploration capability, the random selection of learning sources may cause inefficient exploitation and oscillatory convergence in later stages of the optimization process. To improve search stability and precision, AMS-GRO introduces an Adaptive Elite-Guided Search (AEG) strategy as a unified refinement mechanism.

Within the AEG framework, the informational reference pool available to each prospector is confined exclusively to an elite sub-population, rather than being drawn from the entirety of the swarm. More precisely, at each iteration, an elite exemplar denoted as $X_e^{(t)}$ is stochastically sampled from the uppermost

$$E = \lceil 0.2N \rceil \tag{7}$$

individuals to guide the search direction. The position of the i-th prospector is then updated according to

$$X_i^{(t+1)} = X_e^{(t)} + A \cdot \left(X_i^{(t)} - X_e^{(t)} \right) + \mathcal{N}\left(0, \sigma^2\right) \tag{8}$$

where A is a control coefficient inherited from the GRO framework, and $\mathcal{N}\left(0, \sigma^2\right)$ denotes Gaussian perturbation.

To further enhance robustness, the perturbation intensity is adaptively controlled as

$$\sigma = 0.12 \cdot \left(1 - \tfrac{t}{T}\right)^{1.3} \cdot (ub - lb) \tag{9}$$

where σ denotes the standard deviation of the Gaussian perturbation, t is the current iteration, T is the maximum number of iterations, and ub and lb denote the upper and lower bounds of the search space, respectively. The coefficient 0.12 determines the initial perturbation amplitude, whereas the exponent 1.3 controls its nonlinear decay rate. Therefore, a relatively large perturbation is retained during the early search stage to improve population diversity, while the perturbation gradually approaches zero as the iteration proceeds, thereby facilitating stable local refinement. In addition, a lightweight best-guided refinement is applied to the updated position:

$$\begin{gathered} X_i^{(t+1)} = (1 - \beta) \cdot X_i^{(t+1)} + \beta \cdot X_{best}^{(t)} \\ \beta = 0.2 \cdot \tfrac{t}{T} \end{gathered} \tag{10}$$

where β is the adaptive best-guidance coefficient, $X_{best}^{(t)}$ denotes the best solution found up to iteration t, and $X_i^{(t+1)}$ is the candidate solution generated by the elite-guided update in Equation (8). Since β increases linearly from 0 to 0.2, the influence of the global-best solution is deliberately limited in the early stage and gradually strengthened in the later stage. This design preserves exploration at the beginning of the search while improving convergence accuracy near the end of the optimization process.

By combining elite-guided learning, adaptive perturbation control, and mild best guidance, the AEG strategy enhances local exploitation capability while preserving sufficient exploration.

2.2.3. Success-Rate–Based Adaptive Strategy Selection Mechanism

Different search strategies may exhibit varying effectiveness at different optimization stages. Therefore, a fixed strategy selection scheme cannot guarantee robust performance across diverse problems or iteration phases.

To address this issue, AMS-GRO employs a success-rate–based adaptive strategy selection mechanism to dynamically coordinate multiple search strategies.

At each iteration of AMS-GRO, every prospector constructs one trial solution using a strategy selected from the available pool. For strategy $succ_s$ records the number of trial solutions that replace their corresponding parent solutions during the current iteration. An update is classified as successful when the new solution achieves a better objective value than its parent. The strategy probabilities $\mathbf{P} = [P_1, P_2, P_3]$ are subsequently adjusted as follows:

$$P_s^{(t+1)} = (1-\alpha)P_s^{(t)} + \alpha \cdot \frac{succ_s}{\sum_{k=1}^{3} succ_k} \tag{11}$$

where $\alpha = 0.2$ controls the smoothing between the previous probability and the latest performance of each strategy.

To avoid permanently excluding a strategy because of short-term poor performance, a lower bound $P_s \geq \varepsilon$ is imposed, after which the probability vector is normalized. In this way, AMS-GRO adaptively distributes search resources according to the contribution of each strategy, thereby enhancing its adaptability and robustness.

The adaptive implementation consists of three mutually connected levels. First, the candidate strategy pool contains the current-to-pbest/1 operator, the original GRO operator, and the adaptive elite-guided operator. At the beginning of the optimization process, their selection probabilities are initialized equally, i.e., $P_1^{(0)} = P_2^{(0)} = P_3^{(0)} = 1/3$. For each individual, one operator is sampled according to the categorical distribution defined by $\mathbf{P}(t)$, and only the selected operator generates the trial solution. Consequently, the three operators do not need to be executed simultaneously for every individual.

Second, the internal search intensity of the operators changes with the optimization progress. For the current-to-pbest/1 operator, the size of the elite set decreases from approximately $0.20N$ to $0.05N$, gradually increasing the selection pressure. Its scaling factor $F^{(t)}$ is sampled from the success-history memory and restricted to [0.05,1]. For the adaptive elite-guided operator, the Gaussian perturbation amplitude σ\sigma decreases nonlinearly, whereas the best-guidance coefficient β\beta increases from 0 to 0.2. Thus, exploration is emphasized at the beginning, while local refinement is gradually strengthened.

Third, the allocation of computational resources among the operators is adjusted according to their observed success. A trial is counted as successful only when it improves its parent under greedy selection. At the end of each iteration, the success ratios are combined with the previous probability vector through Equation (11). A lower probability bound ε is then imposed, followed by normalization: $\tilde{P}_s^{(t+1)} = max\left(P_s^{(t+1)}, \varepsilon\right), P_s^{(t+1)} = \frac{\tilde{P}_s^{(t+1)}}{\sum_{k=1}^{3} \tilde{P}_k^{(t+1)}}$.

If no strategy produces a successful update in an iteration, i.e., $\sum_{k=1}^{3} succ_k = 0$, the previous probability vector is retained to avoid division by zero. This feedback process assigns higher probabilities to strategies that have recently generated effective solutions while preserving a nonzero chance of selecting temporarily unsuccessful strategies.

As shown in Figure 2, this mechanism evaluates strategy effectiveness through real-time success rate statistics and dynamically adjusts the probability distribution of strategy selection accordingly.

By incorporating the current-to-pbest/1 mutation strategy, Adaptive Elite-Guided Search, and a success-rate-based adaptive strategy selection mechanism, AMS-GRO improves the convergence stability, search efficiency, and adaptability of the original Gold Rush Optimizer when solving complex optimization problems.

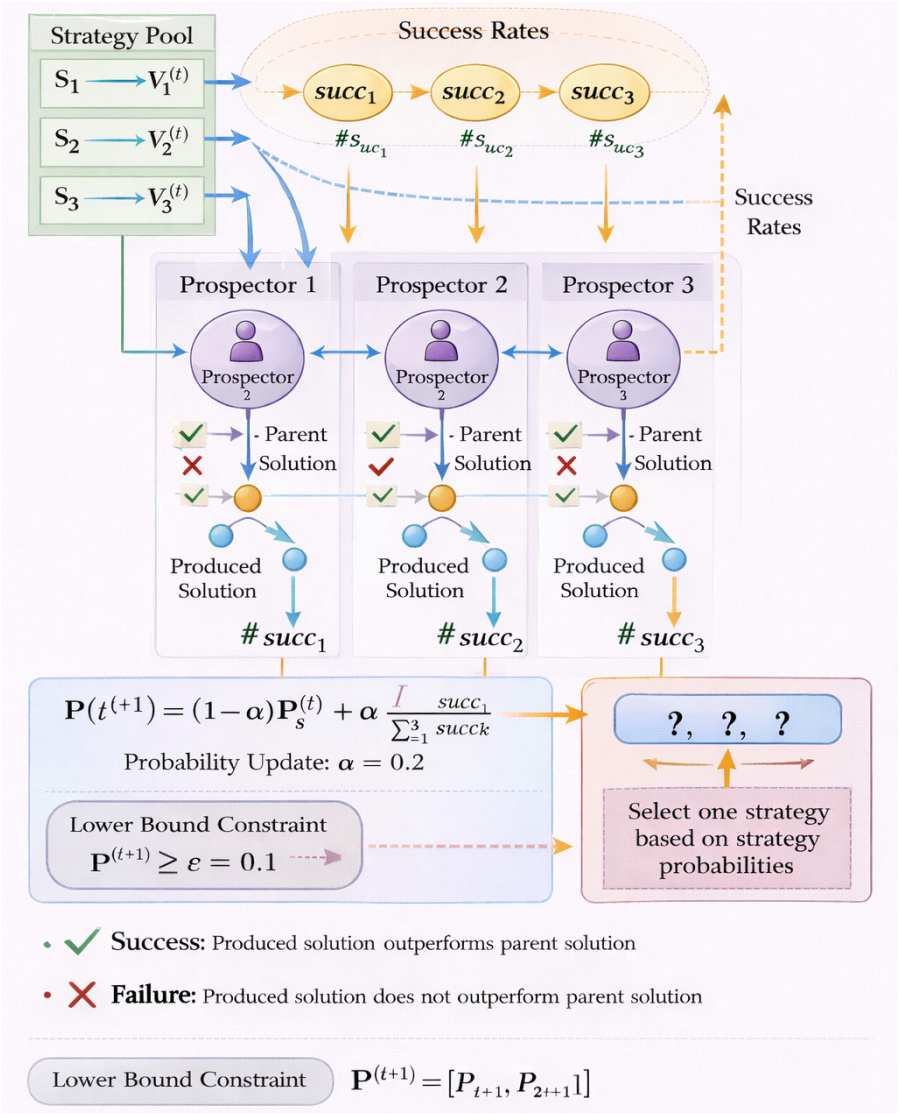


Figure 2. Illustrates the workflow of the success-rate–based adaptive strategy selection mechanism.

A detailed pseudocode for AMS-GRO can be found in Algorithm 1.

2.3. Computational Complexity Analysis

Let N, D, and T denote the population size, problem dimension, and maximum number of iterations, respectively, and let C_f denote the computational cost of one objective-function evaluation. Population initialization requires $O(ND)$ operations and $O\left(NC_f\right)$ for the initial fitness evaluations.

During each iteration, sorting the population according to fitness requires $O(N\log N)$. The current-to-pbestpbest/1 mutation, original GRO search, adaptive elite-guided search, boundary handling, and greedy selection all operate on D-dimensional vectors for N individuals, resulting in $O(ND)$ computational cost. Updating the success counters, adaptive parameter memory, and three-element strategy-probability vector requires $O(N)$ operations and does not change the dominant order. The N candidate solutions generated in each iteration require $O\left(NC_f\right)$ for fitness evaluation. Therefore, the total time complexity of AMS-GRO is $O\left(ND + NC_f + T\left(N\log N + ND + NC_f\right)\right)$.

Algorithm 1: the pseudo-code of the AMS-GRO

Initialize population $X_i^{(0)} \in [lb, ub], i = 1, \ldots, N$; evaluate fitness $f(X_i^{(0)})$.
Set $X_{best} \leftarrow \arg\min_i f(X_i^{(0)})$; initialize archive A.
Initialize adaptive memories and strategy probabilities $\mathbf{P} = [P_1, P_2, P_3]$.
while $t < T$ ***do***
    Sort population according to fitness values; update X_{best}.
    Compute elite sizes p (for pbest) and E (for AEG)
    Reset strategy success counters $succ_s = 0, s = 1, 2, 3$.
    for each individual $X_i^{(t)}$ ***do***
        Sample adaptive control parameters $F^{(t)}$.
        Select a strategy $s \in \{1, 2, 3\}$ according to probability vector **P.**
        if $s = 1$ ***do***
            Update $X_i^{(t+1)}$ according to current-to-pbest/1 mutation.
        else if $s = 2$ ***do***
            Update $X_i^{(t+1)}$ according to Original GRO search.
        else if $s = 3$ ***do***
            Update $X_i^{(t+1)}$ Adaptive Elite-Guided Search.
        end if
        Apply boundary handling to $X_i^{(t+1)}$ and $f\left(X_i^{(t+1)}\right)$
        if $f\left(X_i^{(t+1)}\right) < f(X_i)$ ***do***
            Update $X_i = X_i^{(t+1)}$
        end if
    end for
    Store replaced parent in archive A.
    Increment success counter $succ_s \leftarrow succ_s + 1$.
    Update global best solution X_{best}.
    Update adaptive control memories MF.
    Update strategy probabilities **P**.
    $t = t + 1$
end while
return *the best solution* X_{best}.

When the objective-function evaluation dominates the computational cost, the complexity can be simplified to $O\left(TNC_f\right)$. When only the internal optimization operations are considered, the complexity is $O(T(ND + N\log N))$. The original GRO has an internal complexity of approximately $O(TND)$. Thus, AMS-GRO introduces an additional population-sorting term but preserves the same dominant population–dimension iteration structure.

Regarding space complexity, storing the current population and its fitness values requires $O(ND + N)$. The historical archive contains at most N D-dimensional solutions and therefore requires $O(ND)$ additional space. The adaptive memories, success counters, and strategy probabilities require only $O(N)$ or constant space. Consequently, the overall space complexity of AMS-GRO is $O(ND)$.

3. Performance Assessment

3.1. Reference Algorithms and Experimental Setup

To systematically evaluate the optimization performance of the proposed AMS-GRO, this section conducts empirical experiments on the challenging CEC2017 [26] and CEC2022 benchmark suites. The performance of our approach is rigorously benchmarked against a

diverse array of competing algorithms, comprising both established variants of classical metaheuristics—including Velocity Pausing Particle Swarm Optimization (VPPSO) [27], Improved multi-strategy adaptive Grey Wolf Optimization (IAGWO) [28], and the Enhanced Whale Optimization Algorithm (EWOA) [29]—as well as a set of recently developed algorithms from the past two years, namely Wave Optics Optimization (WOO) [30], Secretary Bird Optimization Algorithm (SBOA) [15], and Kangaroo Escape Optimization (KEO) [31]. In addition, multiple algorithms derived from the Gold Rush Optimizer framework are considered, including the original Gold Rush Optimizer (GRO) [32], Augmented Gold Rush Optimizer (AGRO) [33], and the Quantum-enhanced Gold Rush Optimizer (QHGRO) [34]. The detailed parameter configurations for each compared algorithm are compiled in Table 1 in a systematic manner.

Table 1. Parameter specifications of the algorithms used for comparison.

Algorithms	Parameter Name	Numerical Setting
VPPSO	$c_1, c_2,\ w, \alpha, N_1, N_2$	2, 2, 0.8, [1, 0], 0.15, 0.15
IAGWO	$v_{rand}(t), a,\ \omega, \theta$	$[-20, 20]$, [0, 2], [0.3, 0.9], 0.5
EWOA	$P, \kappa, r,\ l,\ \alpha$	0.2, $1.5 \times N$, $[0,\ 1]$, $[-1,\ 1]$, *Linear reduction from* 2 *to* 1
WOO	λ, d, I_0	5×10^{-4}, 10^{-2}, 1
SBOA	$CF, K,\ R_1, R_2$	$[0,\ 1]$, $\{1,\ 2\}$, $[0,\ 1]$, $[0,\ 1]$
KEO	*Energy threshold*, β	0.5, 0.5
GRO	m	$[0,\ 1]$
AGRO	m	$[0,\ 1]$
QHGRO	m	$[0,\ 1]$
AMS-GRO	$\beta_{base}, c, h, p, exist, \alpha_{initial}, \alpha_{final}, \epsilon, \delta, \gamma$	0.15, 0.15, 0.35, 0.5, 5, 0.9, 0.2, 10^{-6}, 0.9, 0.1

To ensure a robust comparison, all algorithms were tested using a common computational protocol. Each run employed a population of 30 individuals and terminated after 500 iterations. For every algorithm-function combination, 30 independent runs were conducted to account for random fluctuations. The final objective values were summarized using their mean (Ave) and standard deviation (Std). Lower values represent better performance, while bold entries identify the best-performing method.

All experiments were performed on a workstation running Ubuntu 22.04, equipped with a 12th-generation Intel® Core™ i7-12700 processor operating at 2.1 GHz and 32 GB of RAM. All algorithms were implemented and evaluated using MATLAB R2024a.

3.2. Experimental Evaluation on CEC2017 Benchmark Problems

To rigorously examine the optimization efficacy of the proposed Adaptive Multi-Strategy Gold Rush Optimizer (AMS-GRO) across varying problem landscapes—with particular emphasis on its enhancements in global search breadth, local refinement precision, and convergence robustness—a series of numerical trials is undertaken in this section using the 30-dimensional CEC2017 benchmark suite. Through systematic side-by-side comparisons against both classical and cutting-edge optimization techniques, the comprehensive solution quality of AMS-GRO is critically appraised. The detailed findings from these experiments are subsequently reported in Table 2, Figures 3 and 4.

Table 2. Performance comparison of algorithms on the CEC2017 benchmark suite with 30 dimensions.

Function	Metric	VPPSO	IAGWO	EWOA	WOO	SBOA	KEO	GRO	AGRO	QHGRO	AMS-GRO
F1	Ave	6.2964×10^{6}	1.6212×10^{6}	3.0503×10^{9}	4.8053×10^{8}	1.6568×10^{5}	1.2316×10^{5}	1.0852×10^{9}	1.2164×10^{7}	1.6766×10^{4}	1.4426×10^{3}
	Std	1.5589×10^{7}	5.3544×10^{5}	3.5668×10^{9}	1.2688×10^{8}	2.6623×10^{5}	8.3268×10^{4}	1.0186×10^{9}	2.2254×10^{7}	1.1390×10^{4}	2.3107×10^{3}
F2	Ave	2.1237×10^{24}	2.2144×10^{35}	2.3868×10^{37}	2.6086×10^{27}	1.3620×10^{19}	2.4453×10^{21}	7.0437×10^{28}	3.1099×10^{21}	5.3671×10^{17}	1.7622×10^{12}
	Std	6.1240×10^{24}	1.1438×10^{36}	9.7287×10^{37}	1.4001×10^{28}	7.2140×10^{19}	8.3374×10^{21}	2.4416×10^{29}	1.5726×10^{22}	1.4868×10^{18}	6.5381×10^{12}
F3	Ave	4.8465×10^{4}	6.2166×10^{4}	1.8009×10^{5}	4.0569×10^{4}	3.3559×10^{4}	4.5134×10^{4}	5.6172×10^{4}	4.9714×10^{4}	3.6995×10^{4}	7.1305×10^{4}
	Std	9.2484×10^{3}	1.8125×10^{4}	3.3716×10^{4}	9.7521×10^{3}	1.0025×10^{4}	1.9359×10^{4}	1.2253×10^{4}	8.5648×10^{3}	9.7452×10^{3}	5.2217×10^{4}
F4	Ave	5.3390×10^{2}	6.2951×10^{2}	8.6863×10^{2}	5.9236×10^{2}	5.2323×10^{2}	5.2583×10^{2}	6.4163×10^{2}	5.3146×10^{2}	4.9156×10^{2}	4.7926×10^{2}
	Std	2.9826×10^{1}	1.5846×10^{2}	3.3997×10^{2}	2.7620×10^{1}	3.0333×10^{1}	3.4624×10^{1}	7.5343×10^{1}	3.3022×10^{1}	2.5044×10^{1}	3.2515×10^{1}
F5	Ave	6.6161×10^{2}	6.5016×10^{2}	6.9658×10^{2}	6.4091×10^{2}	5.9588×10^{2}	6.3975×10^{2}	6.3128×10^{2}	6.0615×10^{2}	6.2587×10^{2}	5.6799×10^{2}
	Std	3.4335×10^{1}	2.6723×10^{1}	5.4641×10^{1}	1.6883×10^{1}	2.2499×10^{1}	3.8922×10^{1}	2.1591×10^{1}	3.7263×10^{1}	3.1610×10^{1}	1.2125×10^{1}
F6	Ave	6.4025×10^{2}	6.1960×10^{2}	6.4652×10^{2}	6.1117×10^{2}	6.0525×10^{2}	6.2850×10^{2}	6.1753×10^{2}	6.1299×10^{2}	6.1862×10^{2}	6.0004×10^{2}
	Std	8.6758×10^{00}	8.6699×10^{00}	1.2796×10^{1}	1.8313×10^{00}	3.9154×10^{00}	9.8670×10^{00}	5.2352×10^{00}	7.5618×10^{00}	8.0272×10^{00}	8.2906×10^{-2}
F7	Ave	9.3569×10^{2}	9.1908×10^{2}	1.0116×10^{3}	9.3373×10^{2}	8.7499×10^{2}	9.7146×10^{2}	9.1288×10^{2}	8.9565×10^{2}	9.7045×10^{2}	8.1344×10^{2}
	Std	6.3068×10^{1}	4.1929×10^{1}	7.2753×10^{1}	2.2951×10^{1}	5.1937×10^{1}	7.3151×10^{1}	4.2734×10^{1}	4.3440×10^{1}	6.1658×10^{1}	1.2814×10^{1}
F8	Ave	9.2017×10^{2}	9.0700×10^{2}	9.9947×10^{2}	9.2751×10^{2}	8.7886×10^{2}	9.1258×10^{2}	9.1349×10^{2}	9.0403×10^{2}	9.1813×10^{2}	8.6669×10^{2}
	Std	3.3429×10^{1}	1.9143×10^{1}	5.7058×10^{1}	1.2540×10^{1}	2.0155×10^{1}	2.8348×10^{1}	1.9668×10^{1}	2.6769×10^{1}	2.8561×10^{1}	1.1014×10^{1}
F9	Ave	3.1559×10^{3}	4.2649×10^{3}	7.1080×10^{3}	1.8506×10^{3}	2.0170×10^{3}	3.1620×10^{3}	2.0589×10^{3}	1.7123×10^{3}	3.2605×10^{3}	9.4260×10^{2}
	Std	7.3454×10^{2}	1.1116×10^{3}	3.2828×10^{3}	3.5544×10^{2}	5.9939×10^{2}	1.0510×10^{3}	5.3762×10^{2}	4.4674×10^{2}	8.8752×10^{2}	8.4988×10^{1}
F10	Ave	5.0515×10^{3}	4.7734×10^{3}	6.7620×10^{3}	5.9962×10^{3}	4.5712×10^{3}	5.2125×10^{3}	5.1899×10^{3}	6.2458×10^{3}	4.7536×10^{3}	4.6703×10^{3}
	Std	7.2902×10^{2}	6.3159×10^{2}	8.2460×10^{2}	3.3893×10^{2}	6.6709×10^{2}	8.4120×10^{2}	5.3427×10^{2}	6.5898×10^{2}	7.0305×10^{2}	4.3896×10^{2}
F11	Ave	1.4537×10^{3}	1.7485×10^{3}	2.1000×10^{3}	1.3462×10^{3}	1.2307×10^{3}	1.2981×10^{3}	1.4449×10^{3}	1.2723×10^{3}	1.1884×10^{3}	1.1954×10^{3}
	Std	1.4780×10^{2}	6.1144×10^{2}	1.0332×10^{3}	3.5656×10^{1}	4.5006×10^{1}	7.1348×10^{1}	1.3139×10^{2}	4.7302×10^{1}	3.5907×10^{1}	4.7408×10^{1}
F12	Ave	1.8261×10^{7}	5.8972×10^{7}	1.0159×10^{8}	2.6217×10^{7}	1.7388×10^{6}	3.0188×10^{6}	8.4009×10^{6}	1.5338×10^{6}	1.4644×10^{6}	5.5476×10^{4}
	Std	1.4050×10^{7}	1.6935×10^{8}	3.5264×10^{8}	9.9690×10^{6}	1.3308×10^{6}	2.5581×10^{6}	6.7113×10^{6}	1.0724×10^{6}	1.2123×10^{6}	6.0957×10^{4}
F13	Ave	9.7992×10^{4}	5.9268×10^{4}	2.7381×10^{6}	6.2142×10^{5}	2.2011×10^{4}	3.6513×10^{4}	3.6154×10^{4}	7.3827×10^{3}	1.8536×10^{4}	7.5979×10^{3}
	Std	4.3756×10^{4}	2.2447×10^{5}	1.2883×10^{7}	2.5431×10^{5}	2.1516×10^{4}	2.9017×10^{4}	2.2232×10^{4}	6.2601×10^{3}	1.5827×10^{4}	7.8421×10^{3}
F14	Ave	1.0682×10^{5}	5.7712×10^{5}	4.1384×10^{5}	2.2396×10^{3}	4.5806×10^{4}	2.4000×10^{4}	3.4615×10^{4}	4.7089×10^{4}	1.1374×10^{5}	1.5003×10^{4}
	Std	1.1520×10^{5}	4.7538×10^{5}	5.0769×10^{5}	3.9346×10^{2}	3.3296×10^{4}	2.6687×10^{4}	3.3991×10^{4}	9.0472×10^{4}	9.3163×10^{4}	4.8212×10^{4}
F15	Ave	5.2857×10^{4}	7.5468×10^{3}	4.0932×10^{4}	4.5011×10^{4}	1.1252×10^{4}	1.5271×10^{4}	9.6790×10^{3}	5.2281×10^{3}	7.2054×10^{3}	1.7014×10^{3}
	Std	5.3597×10^{4}	7.0386×10^{3}	3.1864×10^{4}	1.4810×10^{4}	1.0279×10^{4}	1.5941×10^{4}	4.9713×10^{3}	4.2137×10^{3}	6.0887×10^{3}	8.5896×10^{1}
F16	Ave	2.8813×10^{3}	3.0875×10^{3}	3.1434×10^{3}	2.7640×10^{3}	2.3885×10^{3}	2.8034×10^{3}	2.5381×10^{3}	2.6750×10^{3}	2.7863×10^{3}	2.3456×10^{3}
	Std	2.9386×10^{2}	4.1983×10^{2}	3.5819×10^{2}	1.7007×10^{2}	3.0618×10^{2}	3.2397×10^{2}	2.7894×10^{2}	2.5246×10^{2}	3.0499×10^{2}	1.8693×10^{2}
F17	Ave	2.2003×10^{3}	2.3685×10^{3}	2.6164×10^{3}	2.1972×10^{3}	1.9724×10^{3}	2.3341×10^{3}	1.9544×10^{3}	1.9906×10^{3}	2.2734×10^{3}	1.8861×10^{3}
	Std	1.9785×10^{2}	2.4505×10^{2}	3.1597×10^{2}	1.5838×10^{2}	1.4024×10^{2}	2.0597×10^{2}	1.4385×10^{2}	1.4368×10^{2}	2.1292×10^{2}	1.0371×10^{2}
F18	Ave	1.0927×10^{6}	1.2135×10^{6}	3.4865×10^{6}	5.6433×10^{4}	9.1890×10^{5}	4.3835×10^{5}	4.8501×10^{5}	4.7864×10^{5}	1.3755×10^{6}	1.3251×10^{5}
	Std	1.5236×10^{6}	1.7454×10^{6}	3.8657×10^{6}	2.3909×10^{4}	1.0642×10^{6}	7.1989×10^{5}	4.0407×10^{5}	5.0829×10^{5}	1.5005×10^{6}	2.0310×10^{5}

Table 2. *Cont.*

Function	Metric	VPPSO	IAGWO	EWOA	WOO	SBOA	KEO	GRO	AGRO	QHGRO	AMS-GRO
F19	Ave	1.9646×10^{6}	7.3551×10^{3}	2.4536×10^{6}	7.5658×10^{4}	8.4167×10^{3}	8.7646×10^{3}	9.0015×10^{3}	5.9874×10^{3}	1.2516×10^{4}	1.9898×10^{3}
	Std	1.1901×10^{6}	6.3149×10^{3}	4.7922×10^{6}	3.9977×10^{4}	9.5527×10^{3}	8.6002×10^{3}	8.9667×10^{3}	4.9538×10^{3}	1.1148×10^{4}	5.9407×10^{1}
F20	Ave	2.5123×10^{3}	2.5870×10^{3}	2.6274×10^{3}	2.4462×10^{3}	2.3148×10^{3}	2.6919×10^{3}	2.3248×10^{3}	2.4179×10^{3}	2.5602×10^{3}	2.3385×10^{3}
	Std	1.6024×10^{2}	1.9997×10^{2}	1.6422×10^{2}	1.1605×10^{2}	1.6773×10^{2}	2.2799×10^{2}	8.2938×10^{1}	1.2782×10^{2}	2.1282×10^{2}	9.3248×10^{1}
F21	Ave	2.4334×10^{3}	2.4516×10^{3}	2.4880×10^{3}	2.4507×10^{3}	2.3790×10^{3}	2.4246×10^{3}	2.4093×10^{3}	2.3885×10^{3}	2.4257×10^{3}	2.3653×10^{3}
	Std	2.9188×10^{1}	3.3660×10^{1}	4.2624×10^{1}	1.5387×10^{1}	2.3168×10^{1}	2.9043×10^{1}	1.7197×10^{1}	2.3873×10^{1}	3.3079×10^{1}	1.1386×10^{1}
F22	Ave	2.9501×10^{3}	3.7021×10^{3}	6.3832×10^{3}	4.6926×10^{3}	2.5611×10^{3}	5.2944×10^{3}	2.8975×10^{3}	2.3263×10^{3}	4.5348×10^{3}	2.4219×10^{3}
	Std	1.5097×10^{3}	2.0142×10^{3}	2.3187×10^{3}	2.4269×10^{3}	9.9376×10^{2}	2.2040×10^{3}	1.2291×10^{3}	1.9440×10^{1}	2.2250×10^{3}	5.6252×10^{2}
F23	Ave	2.8206×10^{3}	3.1313×10^{3}	2.8965×10^{3}	2.8737×10^{3}	2.7408×10^{3}	2.7891×10^{3}	2.7868×10^{3}	2.7573×10^{3}	2.7852×10^{3}	2.7148×10^{3}
	Std	4.7994×10^{1}	1.6813×10^{2}	5.0078×10^{1}	2.5738×10^{1}	2.9538×10^{1}	3.7130×10^{1}	3.2159×10^{1}	3.5844×10^{1}	4.9082×10^{1}	1.2974×10^{1}
F24	Ave	2.9575×10^{3}	3.3241×10^{3}	3.0335×10^{3}	3.0330×10^{3}	2.9069×10^{3}	2.9351×10^{3}	2.9517×10^{3}	2.9219×10^{3}	2.9494×10^{3}	2.8746×10^{3}
	Std	4.3587×10^{1}	1.3268×10^{2}	5.3053×10^{1}	2.7495×10^{1}	2.9085×10^{1}	4.0986×10^{1}	3.0659×10^{1}	2.8242×10^{1}	4.1460×10^{1}	1.1928×10^{1}
F25	Ave	2.9508×10^{3}	2.9604×10^{3}	3.0194×10^{3}	2.9660×10^{3}	2.9147×10^{3}	2.9210×10^{3}	3.0025×10^{3}	2.9352×10^{3}	2.8960×10^{3}	2.8875×10^{3}
	Std	2.6361×10^{1}	4.0458×10^{1}	1.2026×10^{2}	1.9999×10^{1}	2.3483×10^{1}	2.5594×10^{1}	4.2206×10^{1}	2.0838×10^{1}	1.6815×10^{1}	1.7881×10^{00}
F26	Ave	4.8285×10^{3}	4.5963×10^{3}	6.4181×10^{3}	5.7103×10^{3}	3.9424×10^{3}	5.5336×10^{3}	4.7862×10^{3}	4.5416×10^{3}	4.9322×10^{3}	4.1166×10^{3}
	Std	1.1944×10^{3}	1.8247×10^{3}	7.5852×10^{2}	4.0703×10^{2}	9.2541×10^{2}	7.8209×10^{2}	6.9984×10^{2}	9.7524×10^{2}	9.7260×10^{2}	4.6190×10^{2}
F27	Ave	3.3036×10^{3}	3.2733×10^{3}	3.2960×10^{3}	3.2521×10^{3}	3.2309×10^{3}	3.2527×10^{3}	3.2834×10^{3}	3.2607×10^{3}	3.2573×10^{3}	3.2143×10^{3}
	Std	4.0321×10^{1}	2.0849×10^{2}	4.8053×10^{1}	7.9317×10^{00}	1.2598×10^{1}	2.2889×10^{1}	2.3015×10^{1}	1.8066×10^{1}	2.5261×10^{1}	8.0565×10^{00}
F28	Ave	3.3283×10^{3}	3.3870×10^{3}	3.4786×10^{3}	3.4047×10^{3}	3.2662×10^{3}	3.2933×10^{3}	3.4005×10^{3}	3.3208×10^{3}	3.2332×10^{3}	3.1981×10^{3}
	Std	4.1365×10^{1}	9.3716×10^{1}	2.2870×10^{2}	4.3789×10^{1}	3.0825×10^{1}	3.9055×10^{1}	6.4235×10^{1}	3.8763×10^{1}	2.2740×10^{1}	3.7151×10^{1}
F29	Ave	4.2752×10^{3}	3.9377×10^{3}	4.3850×10^{3}	3.9206×10^{3}	3.6761×10^{3}	4.2870×10^{3}	3.8749×10^{3}	3.8509×10^{3}	3.9518×10^{3}	3.6649×10^{3}
	Std	2.9075×10^{2}	2.3473×10^{2}	3.5525×10^{2}	1.5871×10^{2}	1.4648×10^{2}	2.6762×10^{2}	1.3461×10^{2}	1.7900×10^{2}	2.2239×10^{2}	1.0662×10^{2}
F30	Ave	7.4091×10^{6}	5.9215×10^{4}	1.4316×10^{6}	5.1299×10^{5}	2.1572×10^{4}	1.0274×10^{5}	2.3936×10^{5}	1.4971×10^{4}	1.0869×10^{4}	7.6777×10^{3}
	Std	4.7133×10^{6}	2.9073×10^{5}	4.6806×10^{6}	2.0230×10^{5}	1.2474×10^{4}	1.1396×10^{5}	2.6461×10^{5}	5.4429×10^{3}	3.2860×10^{3}	1.8302×10^{3}

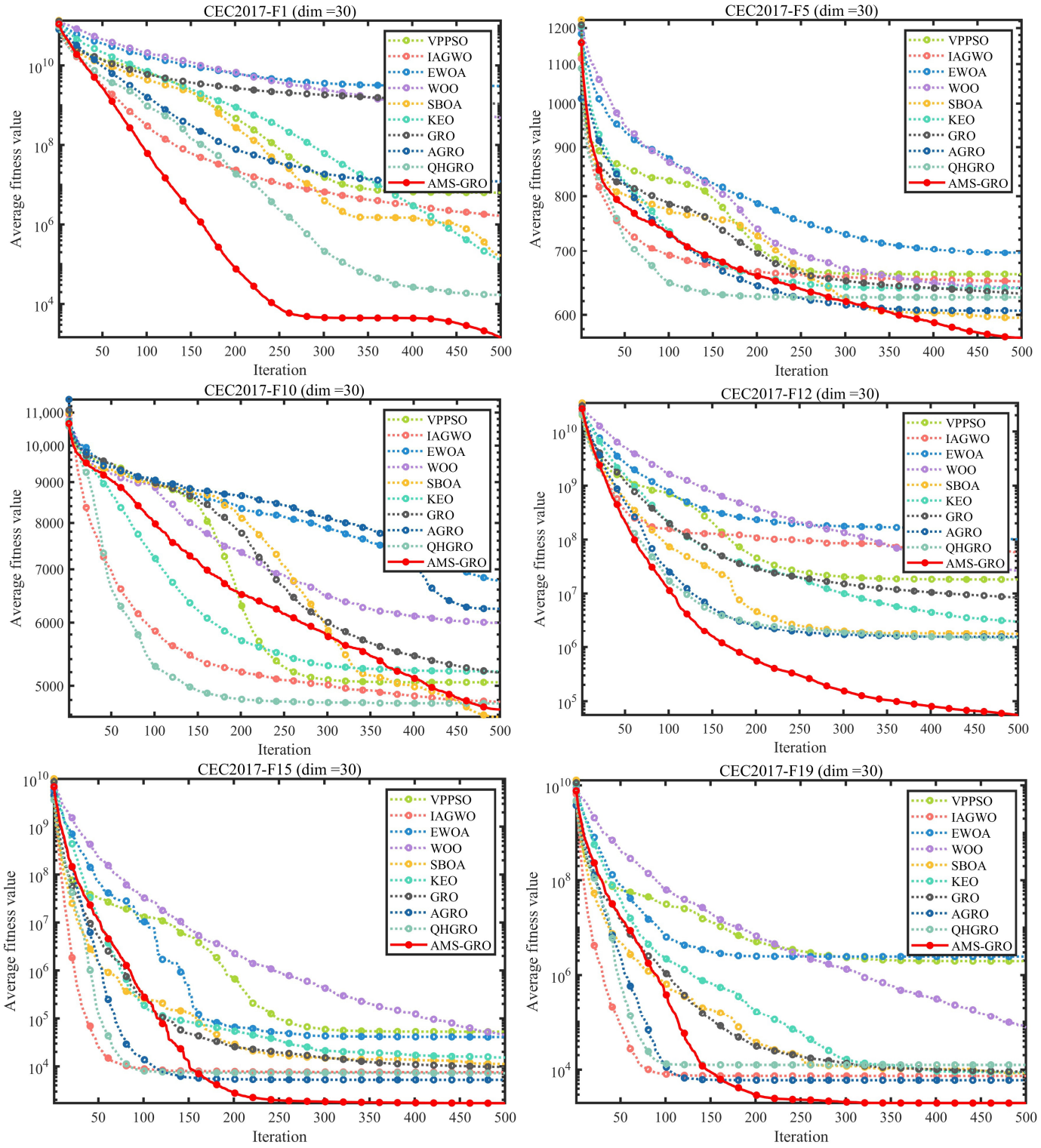


Figure 3. *Cont.*

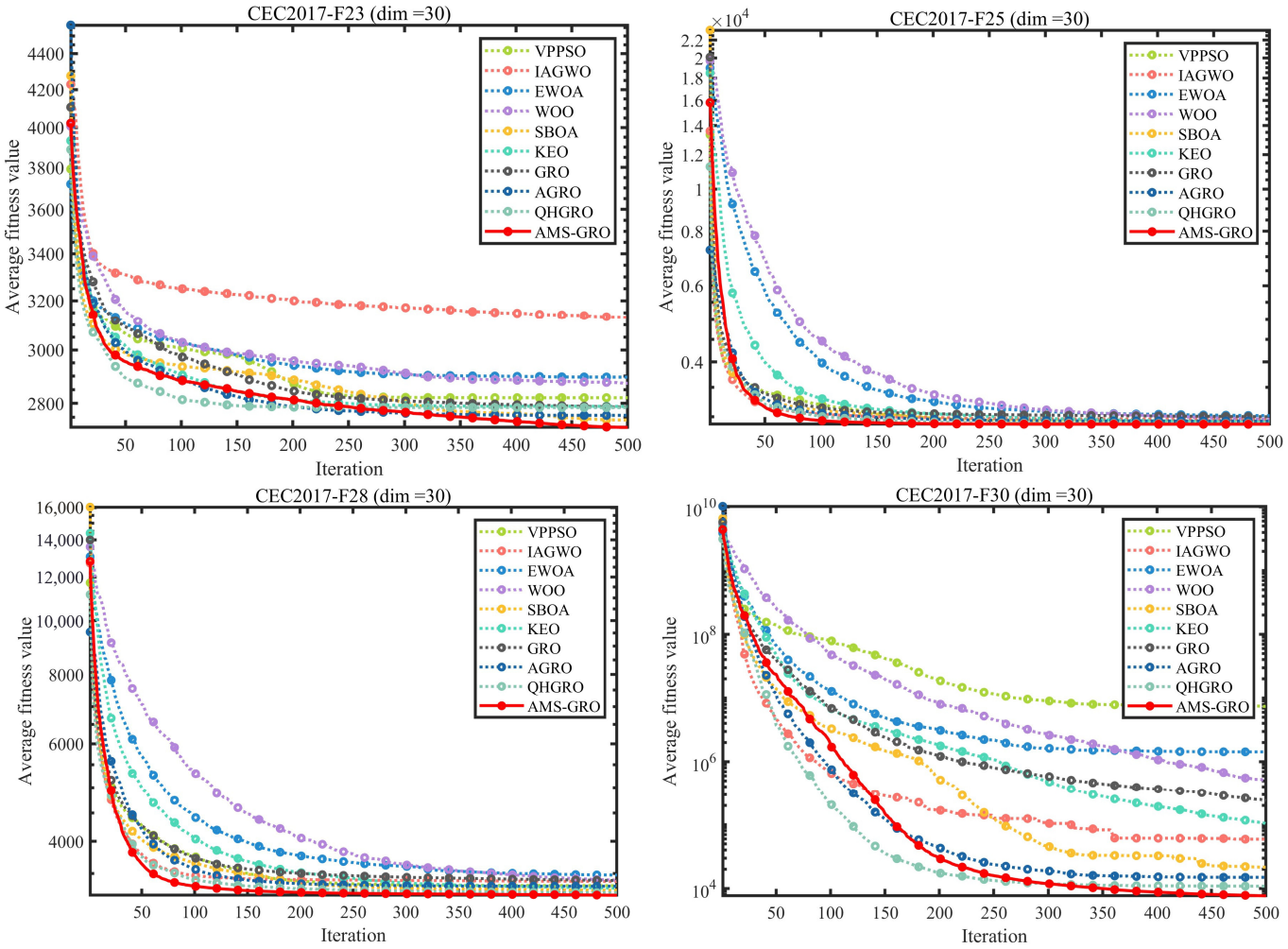


Figure 3. Convergence curves obtained on the CEC2017 benchmark functions.

The optimization outcomes of AMS-GRO relative to its competing counterparts are systematically summarized in Table 2, which reports the average fitness values (Ave) along with their corresponding standard deviations (Std) for both unimodal (F1–F5) and multimodal (F6–F10) test functions drawn from the 30-dimensional CEC2017 benchmark collection. For unimodal functions, AMS-GRO demonstrates outstanding global optimization accuracy and stability. Specifically, for F1, a representative Sphere function, AMS-GRO achieves an Ave of only 1.4426×10^3, which is far lower than that of the original GRO (1.0852×10^9), QHGRO (1.6766×10^4), and even AGRO within the same framework (1.2164×10^7), corresponding to a reduction of 99.99%. The Std is controlled at 2.3107×10^3, indicating that AMS-GRO can accurately locate the global optimum region in smooth unimodal landscapes. For F2, a high-dimensional unimodal function, the Ave of AMS-GRO (1.7622×10^{12}) is more than 20 orders of magnitude lower than those of IAGWO (2.2144×10^{35}) and EWOA (2.3868×10^{37}); even compared with the relatively competitive QHGRO (5.3671×10^{17}), AMS-GRO improves the optimization accuracy by 5 orders of magnitude. Although the Ave of AMS-GRO on F3 (7.1305×10^4) is slightly higher than those of SBOA (3.3559×10^4) and QHGRO (3.6995×10^4), an examination of the Std (5.2217×10^4) suggests that the fluctuation mainly arises from the intrinsic local characteristics of the function, while AMS-GRO still maintains a relatively stable search behavior. For F4, AMS-GRO achieves the lowest Ave among all algorithms (4.7926×10^2), which is 2.5% lower than the second-best QHGRO (4.9156×10^2); the Std of 3.2515×10^1 is only slightly higher than that of QHGRO, reflecting precise local refinement capability. On F5,

the Ave of AMS-GRO (5.6799×10^2) is reduced by 4.7% and 6.3% compared with SBOA (5.9588×10^2) and AGRO (6.0615×10^2), respectively, while the Std (1.2125×10^1) is only 45.4% of that of IAGWO (2.6723×10^1), further confirming its advantage in achieving both high accuracy and stability on unimodal functions.

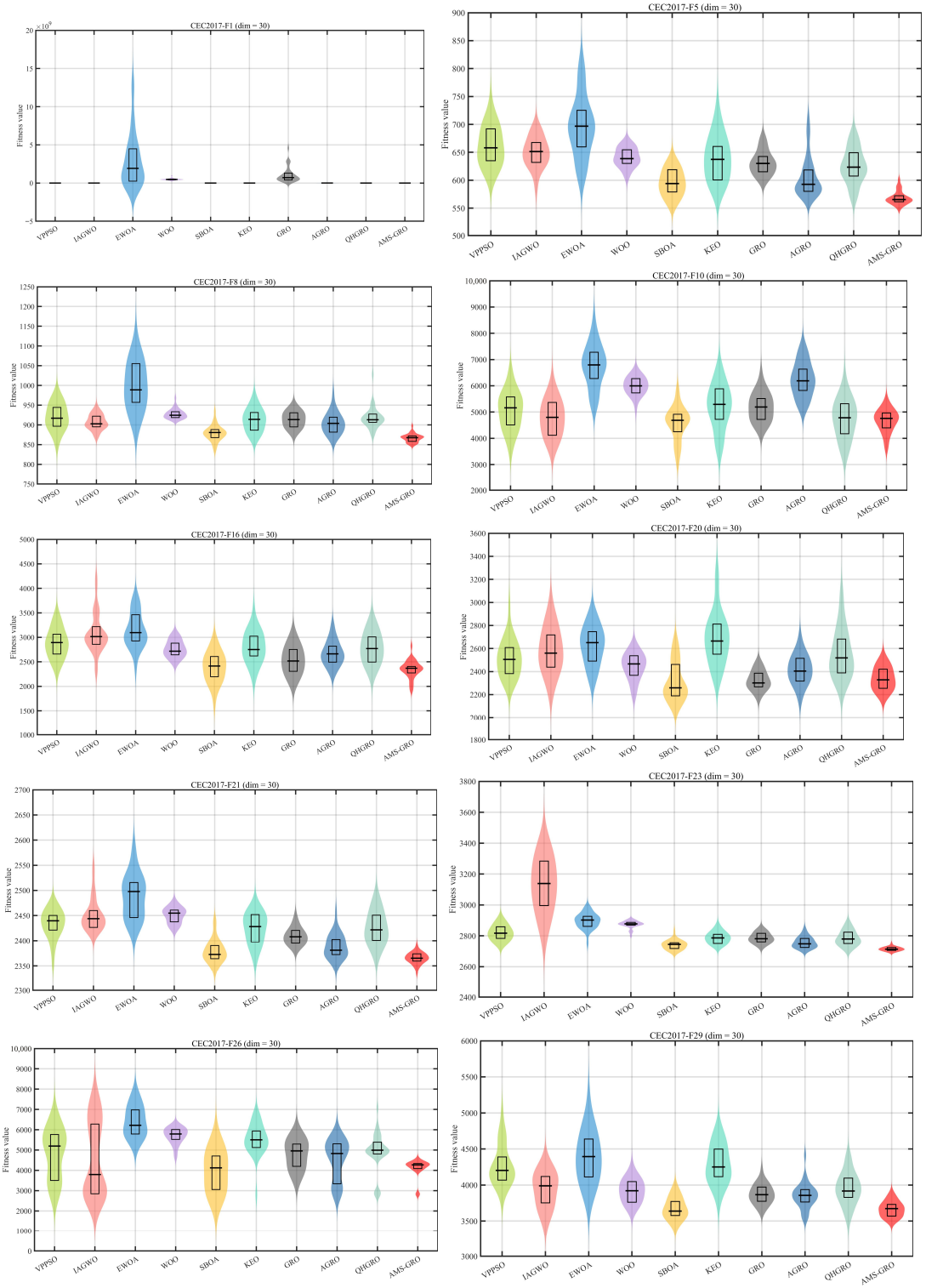


Figure 4. Boxplots of the optimization results obtained by different algorithms on the CEC2017 benchmark functions.

For multimodal functions, AMS-GRO exhibits excellent capability in escaping local optima by balancing population diversity and local exploitation. On F6, the Ave of 6.0004×10^{2} is close to the theoretical optimum, and the Std is only 8.2906×10^{-2}, representing the strongest stability among all algorithms and a reduction of 97.9% compared with the Std of SBOA (3.9154×10^{00}). For F7, the Ave of AMS-GRO (8.1344×10^{2}) is 7.0% lower than that of the best-performing SBOA (8.7499×10^{2}), while the Std of 1.2814×10^{1} is only 30.6% of that of IAGWO (4.1929×10^{1}), indicating its effectiveness in avoiding premature convergence in multimodal landscapes. On F8, AMS-GRO achieves an Ave of 8.6669×10^{2}, which is 1.4% lower than that of SBOA (8.7886×10^{2}), and a Std of 1.1014×10^{1}, which is much smaller than those of VPPSO (3.3429×10^{1}) and EWOA (5.7058×10^{1}), implying more concentrated search results. For F9, a complex multimodal function, the Ave of AMS-GRO (9.4260×10^{2}) is reduced by 45.0% compared with the second-best AGRO (1.7123×10^{3}), and the Std of 8.4988×10^{1} is only 7.7% of that of IAGWO (1.1116×10^{3}), demonstrating strong local exploitation and global coordination capability. Although the Ave of AMS-GRO on F10 (4.6703×10^{3}) is slightly higher than that of SBOA (4.5712×10^{3}), its Std (4.3896×10^{2}) is lower than those of SBOA (6.6709×10^{2}) and IAGWO (6.3159×10^{2}), indicating reduced sensitivity to randomness and a more robust search process.

Figure 3 presents the mean convergence trajectories of the compared algorithms on ten representative 30-dimensional CEC2017 functions. On F1, AMS-GRO exhibits a sustained decline during the first approximately 260 iterations rather than becoming stable within the first 100 iterations. After entering a relatively low-fitness region, the algorithm continues to produce a further improvement near the end of the search and obtains the lowest final fitness value among the compared methods. On F5, the AMS-GRO curve decreases continuously throughout most of the optimization process and reaches approximately 5.7×10^{2} at the final iteration, which is consistent with the mean value reported in Table 2.

On F10, AMS-GRO does not dominate throughout the entire process. Its convergence is relatively gradual in the middle stage, but the curve continues to decline after several competing methods have entered plateaus. Its final value is competitive with those of SBOA, IAGWO, and QHGRO, consistent with the result in Table 2 that AMS-GRO ranks second in terms of the mean objective value on this function. On F12, AMS-GRO maintains a steady decline over the complete search process and ultimately reaches a markedly lower fitness region than the competing algorithms. For F15 and F19, the proposed method shows rapid improvement during the early and middle stages and reaches lower final fitness levels, indicating effective refinement on these hybrid and composition landscapes.

For F23 and F28, the differences between the algorithms are smaller than those observed on F1, F12, and F19. Nevertheless, AMS-GRO generally maintains a lower trajectory in the later stage and obtains the lowest final mean values. On F25, most algorithms converge rapidly during the early stage, while AMS-GRO reaches a competitive region earlier and retains a small final advantage. On F30, AMS-GRO continues to improve after approximately 200 iterations and obtains one of the lowest final convergence levels. Overall, the curves indicate that the advantage of AMS-GRO is not based on uniformly faster convergence at every iteration; instead, it arises from its ability to preserve continued improvement on several functions after competing algorithms begin to stagnate.

The boxplots in Figure 4 provide complementary performance validation from a statistical distribution perspective. For unimodal functions F1, F4, and F5, AMS-GRO exhibits significantly lower box heights than other algorithms, with medians close to the optimal values and no outliers. In the boxplot of F1, the distribution of AMS-GRO is concentrated within the range of 1.0×10^{3}–3.5×10^{3}, whereas the boxes of VPPSO and EWOA span more than 1.0×10^{7} with numerous outliers. For F5, the box of AMS-GRO is almost entirely concentrated within 550–580, with a median of approximately 568, which is

far lower than the medians of other algorithms (all above 600). On multimodal functions F6, F7, and F9, AMS-GRO also shows "narrow and low" box characteristics. For F6, the box height is close to zero, and the median nearly coincides with the theoretical optimum. For F9, the box is concentrated within 900–1000, while the boxes of IAGWO and EWOA span more than 2000 with frequent outliers. These statistical characteristics indicate that AMS-GRO achieves much higher consistency across multiple independent runs than the comparison algorithms, is less affected by random initialization and function properties, and exhibits superior robustness.

Taken together, the numerical results in Table 2, convergence profiles in Figure 3, and statistical distributions in Figure 4 demonstrate the overall competitiveness of AMS-GRO. The current-to-pbest/1 mutation strategy, adaptive elite-guided search, and success-rate-based strategy selection jointly promote fast and accurate convergence on unimodal functions. For multimodal problems, these components help preserve an effective balance between global exploration and local exploitation. Consequently, AMS-GRO achieves higher optimization accuracy, faster convergence, and greater stability than the original GRO and the other competing methods. These complementary findings provide consistent evidence for the effectiveness of the three proposed enhancement strategies.

3.3. Experimental Evaluation on CEC2022 Benchmark Problems

To further scrutinize the generalization capability of AMS-GRO across varying dimensionalities and distinct benchmark collections, additional experiments are carried out in this section on the 10-dimensional and 20-dimensional test instances from the CEC2022 suite. By conducting side-by-side performance assessments against a range of reference techniques on both unimodal and multimodal problem types, the broad applicability and practical efficacy of the proposed enhancements are thoroughly examined. Tables 3 and 4 and Figures 5 and 6 provide a comprehensive summary of the experimental results.

The Ave and Std values obtained on the 10- and 20-dimensional CEC2022 tests are reported in Tables 3 and 4, respectively. Overall, AMS-GRO remained highly competitive across different dimensional settings and problem structures. Its performance in the 10-dimensional experiments was particularly strong on F3 and F5. For F3, AMS-GRO achieved an Ave of 6.0000×10^{2}, equal to the theoretical optimum, with a Std of 2.0244×10^{-5}. Among the compared methods, AMS-GRO was the only algorithm that reached the reported optimum on this function. On F5, it obtained an Ave of 9.0000×10^{2} with zero standard deviation. This result outperformed the second-ranked SBOA (9.0086×10^{2}) and showed complete consistency across the independent runs. AMS-GRO also performed strongly on the more complex benchmark functions. On the hybrid function F6, its Ave of 1.8030×10^{3} was 15.7% lower than that of AGRO (2.1393×10^{3}). Its Std of 3.1394×10^{00} was also lower than those of the competing methods, indicating high run-to-run stability. For the composition function F11, AMS-GRO obtained an Ave of 2.6151×10^{3}. This value was approximately 1.1% lower than those achieved by QHGRO (2.6434×10^{3}) and AGRO (2.6451×10^{3}). Moreover, its Std of 4.5966×10^{1} corresponded to only 27.9% of the value produced by VPPSO (1.6453×10^{2}). These results demonstrate that AMS-GRO maintains favorable optimization accuracy and stability across functions with different landscape characteristics.

Table 3. Performance comparison of algorithms on the CEC2022 benchmark suite with 10 dimensions.

Function	Metric	VPPSO	IAGWO	EWOA	WOO	SBOA	KEO	GRO	AGRO	QHGRO	AMS-GRO
F1	Ave	3.2586×10^{2}	3.5629×10^{2}	3.7952×10^{3}	3.0000×10^{2}	3.0220×10^{2}	3.0020×10^{2}	5.3799×10^{2}	3.2358×10^{2}	3.0010×10^{2}	3.0619×10^{2}
	Std	4.7880×10^{1}	8.3806×10^{1}	2.4970×10^{3}	1.4122×10^{-5}	8.5526×10^{00}	6.6513×10^{-1}	2.7042×10^{2}	4.0759×10^{1}	2.4046×10^{-1}	2.4008×10^{1}
F2	Ave	4.0930×10^{2}	4.3911×10^{2}	4.1562×10^{2}	4.0727×10^{2}	4.0519×10^{2}	4.0768×10^{2}	4.0191×10^{2}	4.0399×10^{2}	4.0855×10^{2}	4.0184×10^{2}
	Std	1.5538×10^{1}	3.6772×10^{1}	2.2311×10^{1}	1.3396×10^{1}	3.4900×10^{00}	1.1769×10^{1}	3.5590×10^{00}	1.2883×10^{1}	1.7966×10^{1}	2.9291×10^{00}
F3	Ave	6.0635×10^{2}	6.0034×10^{2}	6.0632×10^{2}	6.0093×10^{2}	6.0001×10^{2}	6.0388×10^{2}	6.0036×10^{2}	6.0006×10^{2}	6.0162×10^{2}	6.0000×10^{2}
	Std	3.7158×10^{00}	2.9822×10^{-1}	6.5851×10^{00}	1.4564×10^{00}	4.7305×10^{-2}	4.5581×10^{00}	2.8850×10^{-1}	1.3022×10^{-1}	3.4215×10^{00}	2.0244×10^{-5}
F4	Ave	8.1867×10^{2}	8.2266×10^{2}	8.2777×10^{2}	8.2073×10^{2}	8.1118×10^{2}	8.1897×10^{2}	8.1057×10^{2}	8.1352×10^{2}	8.2364×10^{2}	8.0588×10^{2}
	Std	5.1645×10^{00}	7.7686×10^{00}	1.0920×10^{1}	7.5098×10^{00}	4.1698×10^{00}	8.6385×10^{00}	3.3848×10^{00}	5.4567×10^{00}	8.1564×10^{00}	2.2176×10^{00}
F5	Ave	9.1001×10^{2}	9.2413×10^{2}	9.7030×10^{2}	9.4441×10^{2}	9.0086×10^{2}	9.3739×10^{2}	9.0530×10^{2}	9.0190×10^{2}	1.0123×10^{3}	9.0000×10^{2}
	Std	1.1272×10^{1}	5.0527×10^{1}	7.0714×10^{1}	4.9750×10^{1}	2.8993×10^{00}	5.0562×10^{1}	1.8454×10^{1}	3.0940×10^{00}	1.4243×10^{2}	0.0000×10^{00}
F6	Ave	4.0253×10^{3}	2.7418×10^{3}	5.3209×10^{3}	2.1478×10^{3}	3.7399×10^{3}	2.6709×10^{3}	2.7477×10^{3}	2.1393×10^{3}	3.3770×10^{3}	1.8030×10^{3}
	Std	1.9713×10^{3}	8.5678×10^{2}	2.0156×10^{3}	1.1240×10^{3}	1.5965×10^{3}	1.1440×10^{3}	1.1151×10^{3}	2.9488×10^{2}	1.4320×10^{3}	3.1394×10^{00}
F7	Ave	2.0324×10^{3}	2.0152×10^{3}	2.0296×10^{3}	2.0259×10^{3}	2.0155×10^{3}	2.0328×10^{3}	2.0161×10^{3}	2.0160×10^{3}	2.0265×10^{3}	2.0020×10^{3}
	Std	1.1197×10^{1}	8.8038×10^{00}	1.3300×10^{1}	9.3583×10^{00}	8.7767×10^{00}	1.5932×10^{1}	1.0713×10^{1}	8.8077×10^{00}	8.7520×10^{00}	2.0394×10^{00}
F8	Ave	2.2252×10^{3}	2.2214×10^{3}	2.2245×10^{3}	2.2217×10^{3}	2.2193×10^{3}	2.2273×10^{3}	2.2188×10^{3}	2.2168×10^{3}	2.2213×10^{3}	2.2140×10^{3}
	Std	3.0281×10^{00}	6.8343×10^{-1}	3.6582×10^{00}	2.3983×10^{1}	6.6805×10^{00}	2.1983×10^{1}	5.9944×10^{00}	7.8909×10^{00}	1.1214×10^{00}	7.1558×10^{00}
F9	Ave	2.5462×10^{3}	2.5259×10^{3}	2.5328×10^{3}	2.5342×10^{3}	2.5293×10^{3}	2.5293×10^{3}	2.5299×10^{3}	2.5300×10^{3}	2.5293×10^{3}	2.5293×10^{3}
	Std	4.0595×10^{1}	3.0428×10^{1}	7.5623×10^{00}	2.6826×10^{1}	2.5333×10^{-13}	9.4830×10^{-7}	2.0030×10^{00}	3.8172×10^{00}	1.4484×10^{-3}	0.0000×10^{00}
F10	Ave	2.5316×10^{3}	2.5772×10^{3}	2.5010×10^{3}	2.5615×10^{3}	2.5665×10^{3}	2.5656×10^{3}	2.5479×10^{3}	2.5082×10^{3}	2.5705×10^{3}	2.5144×10^{3}
	Std	5.2130×10^{1}	6.3469×10^{1}	2.7662×10^{-1}	6.3056×10^{1}	5.4923×10^{1}	7.6071×10^{1}	5.5307×10^{1}	2.9590×10^{1}	5.7955×10^{1}	3.6433×10^{1}
F11	Ave	2.6929×10^{3}	2.7253×10^{3}	2.7125×10^{3}	2.7330×10^{3}	2.7339×10^{3}	2.7602×10^{3}	2.7146×10^{3}	2.6451×10^{3}	2.6434×10^{3}	2.6151×10^{3}
	Std	1.6453×10^{2}	1.4495×10^{2}	1.0042×10^{2}	1.5445×10^{2}	1.7025×10^{2}	1.5390×10^{2}	1.3505×10^{2}	8.9511×10^{1}	9.8973×10^{1}	4.5966×10^{1}
F12	Ave	2.8637×10^{3}	2.8873×10^{3}	2.8660×10^{3}	2.8643×10^{3}	2.8635×10^{3}	2.8654×10^{3}	2.8666×10^{3}	2.8671×10^{3}	2.8676×10^{3}	2.8645×10^{3}
	Std	1.5892×10^{00}	2.4895×10^{1}	2.1754×10^{00}	2.8584×10^{00}	1.6343×10^{00}	2.0636×10^{00}	3.0265×10^{00}	1.9072×10^{00}	3.6055×10^{00}	7.8487×10^{-1}

Table 4. Performance comparison of algorithms on the CEC2022 benchmark suite with 20 dimensions.

Function	Metric	VPPSO	IAGWO	EWOA	WOO	SBOA	KEO	GRO	AGRO	QHGRO	AMS-GRO
F1	Ave	6.4672×10^{3}	1.2603×10^{4}	3.9462×10^{4}	9.1493×10^{3}	3.7975×10^{3}	4.8222×10^{3}	1.1243×10^{4}	9.3237×10^{3}	3.8651×10^{3}	1.0652×10^{4}
	Std	2.7122×10^{3}	5.4340×10^{3}	1.0345×10^{4}	4.5929×10^{3}	2.3084×10^{3}	3.5017×10^{3}	2.5651×10^{3}	4.0240×10^{3}	1.5672×10^{3}	1.1982×10^{4}
F2	Ave	4.7509×10^{2}	4.9794×10^{2}	4.9728×10^{2}	4.6122×10^{2}	4.6762×10^{2}	4.6620×10^{2}	4.9052×10^{2}	4.6806×10^{2}	4.5435×10^{2}	4.4932×10^{2}
	Std	2.5885×10^{1}	3.7163×10^{1}	5.4335×10^{1}	6.1484×10^{00}	2.5649×10^{1}	3.0458×10^{1}	2.2714×10^{1}	1.8074×10^{1}	1.1619×10^{1}	1.1026×10^{1}
F3	Ave	6.2708×10^{2}	6.0938×10^{2}	6.2775×10^{2}	6.0547×10^{2}	6.0110×10^{2}	6.1653×10^{2}	6.0842×10^{2}	6.0222×10^{2}	6.0725×10^{2}	6.0000×10^{2}
	Std	1.2855×10^{1}	5.3113×10^{00}	1.1252×10^{1}	1.8754×10^{00}	1.4856×10^{00}	1.1696×10^{1}	3.3107×10^{00}	1.2596×10^{00}	6.7304×10^{00}	9.9155×10^{-3}
F4	Ave	8.5864×10^{2}	8.6991×10^{2}	9.0288×10^{2}	8.6614×10^{2}	8.4122×10^{2}	8.6506×10^{2}	8.4527×10^{2}	8.4454×10^{2}	8.6308×10^{2}	8.3145×10^{2}
	Std	1.3052×10^{1}	1.8490×10^{1}	2.8714×10^{1}	1.0272×10^{1}	1.2835×10^{1}	2.2443×10^{1}	1.0357×10^{1}	1.2033×10^{1}	2.2119×10^{1}	7.9430×10^{00}
F5	Ave	1.4109×10^{3}	1.9831×10^{3}	2.4507×10^{3}	1.1158×10^{3}	1.0459×10^{3}	1.3534×10^{3}	1.0776×10^{3}	9.6570×10^{2}	1.6176×10^{3}	9.0357×10^{2}
	Std	3.0671×10^{2}	4.7881×10^{2}	7.8991×10^{2}	1.4776×10^{2}	1.9321×10^{2}	2.5408×10^{2}	1.3903×10^{2}	3.8953×10^{1}	3.6689×10^{2}	1.0520×10^{1}
F6	Ave	4.7214×10^{3}	4.9004×10^{3}	1.1360×10^{6}	4.5716×10^{5}	7.2751×10^{3}	8.6888×10^{3}	1.0844×10^{4}	4.1641×10^{3}	4.2938×10^{3}	2.2478×10^{3}
	Std	4.4122×10^{3}	2.6794×10^{3}	5.3473×10^{6}	2.1022×10^{5}	7.2110×10^{3}	7.3636×10^{3}	1.4829×10^{4}	3.1276×10^{3}	3.0540×10^{3}	1.2805×10^{3}
F7	Ave	2.1027×10^{3}	2.0791×10^{3}	2.1230×10^{3}	2.0676×10^{3}	2.0521×10^{3}	2.0969×10^{3}	2.0638×10^{3}	2.0646×10^{3}	2.0855×10^{3}	2.0462×10^{3}
	Std	3.3625×10^{1}	5.2072×10^{1}	3.6315×10^{1}	1.0042×10^{1}	2.8473×10^{1}	3.8349×10^{1}	1.6181×10^{1}	1.7270×10^{1}	4.2811×10^{1}	1.0739×10^{1}
F8	Ave	2.2569×10^{3}	2.2490×10^{3}	2.2957×10^{3}	2.2338×10^{3}	2.2277×10^{3}	2.2657×10^{3}	2.2280×10^{3}	2.2295×10^{3}	2.2339×10^{3}	2.2240×10^{3}
	Std	4.8038×10^{1}	4.8691×10^{1}	6.8397×10^{1}	4.2005×10^{00}	4.0729×10^{00}	5.5097×10^{1}	2.3106×10^{00}	3.0469×10^{00}	3.0318×10^{1}	9.3091×10^{-1}
F9	Ave	2.5103×10^{3}	2.5111×10^{3}	2.5212×10^{3}	2.4820×10^{3}	2.4808×10^{3}	2.4809×10^{3}	2.4858×10^{3}	2.4818×10^{3}	2.4812×10^{3}	2.4808×10^{3}
	Std	2.7887×10^{1}	4.6629×10^{1}	3.8660×10^{1}	4.5086×10^{-1}	5.1573×10^{-2}	3.2653×10^{-1}	3.2488×10^{00}	9.9167×10^{-1}	3.5931×10^{-1}	7.5184×10^{-9}
F10	Ave	3.3728×10^{3}	2.7342×10^{3}	2.5922×10^{3}	2.7895×10^{3}	2.6230×10^{3}	3.8866×10^{3}	2.7564×10^{3}	2.5667×10^{3}	2.9173×10^{3}	2.5118×10^{3}
	Std	1.1177×10^{3}	2.3584×10^{2}	3.1060×10^{2}	2.9486×10^{2}	2.2250×10^{2}	8.8214×10^{2}	5.1205×10^{2}	1.9284×10^{2}	3.8954×10^{2}	6.1610×10^{1}
F11	Ave	2.9215×10^{3}	2.9892×10^{3}	3.8574×10^{3}	3.3558×10^{3}	2.9410×10^{3}	2.9089×10^{3}	3.1788×10^{3}	2.9257×10^{3}	2.9101×10^{3}	2.9167×10^{3}
	Std	8.6341×10^{1}	3.4347×10^{2}	5.6116×10^{2}	7.6111×10^{1}	3.1583×10^{2}	1.4431×10^{2}	1.1823×10^{2}	1.0661×10^{2}	7.1189×10^{1}	3.7905×10^{1}
F12	Ave	2.9917×10^{3}	2.9462×10^{3}	2.9970×10^{3}	2.9554×10^{3}	2.9512×10^{3}	2.9877×10^{3}	2.9834×10^{3}	2.9791×10^{3}	2.9656×10^{3}	2.9457×10^{3}
	Std	4.7141×10^{1}	1.0419×10^{2}	3.4169×10^{1}	3.7016×10^{00}	1.1959×10^{1}	5.7006×10^{1}	1.7349×10^{1}	1.7495×10^{1}	1.6619×10^{1}	9.8049×10^{00}

When extended to the higher-dimensional 20-dimensional scenario, the advantages of AMS-GRO become even more pronounced. For the unimodal function F3, the Ave remains at the theoretical optimum of 6.0000×10^2, with a negligible Std of 9.9155×10^{-3}. For F5, the Ave of AMS-GRO (9.0357×10^2) is reduced by 6.4% compared with AGRO (9.6570×10^2), and the Std of 1.0520×10^1 is only 2.2% of that of IAGWO (4.7881×10^2), highlighting its remarkable stability in high-dimensional spaces. For multimodal functions, AMS-GRO continues to outperform the comparison algorithms. On F6, the Ave of 2.2478×10^3 is reduced by 47.7% compared with QHGRO (4.2938×10^3) and by 45.9% compared with AGRO (4.1641×10^3), indicating that AMS-GRO can accurately locate the global optimum even under complex high-dimensional multimodal conditions. On F10, AMS-GRO achieves an Ave of 2.5118×10^3, which is 2.1% lower than that of the second-best AGRO (2.5667×10^3), while the Std of 6.1610×10^1 is the lowest among all algorithms, reflecting strong resistance to random disturbances.

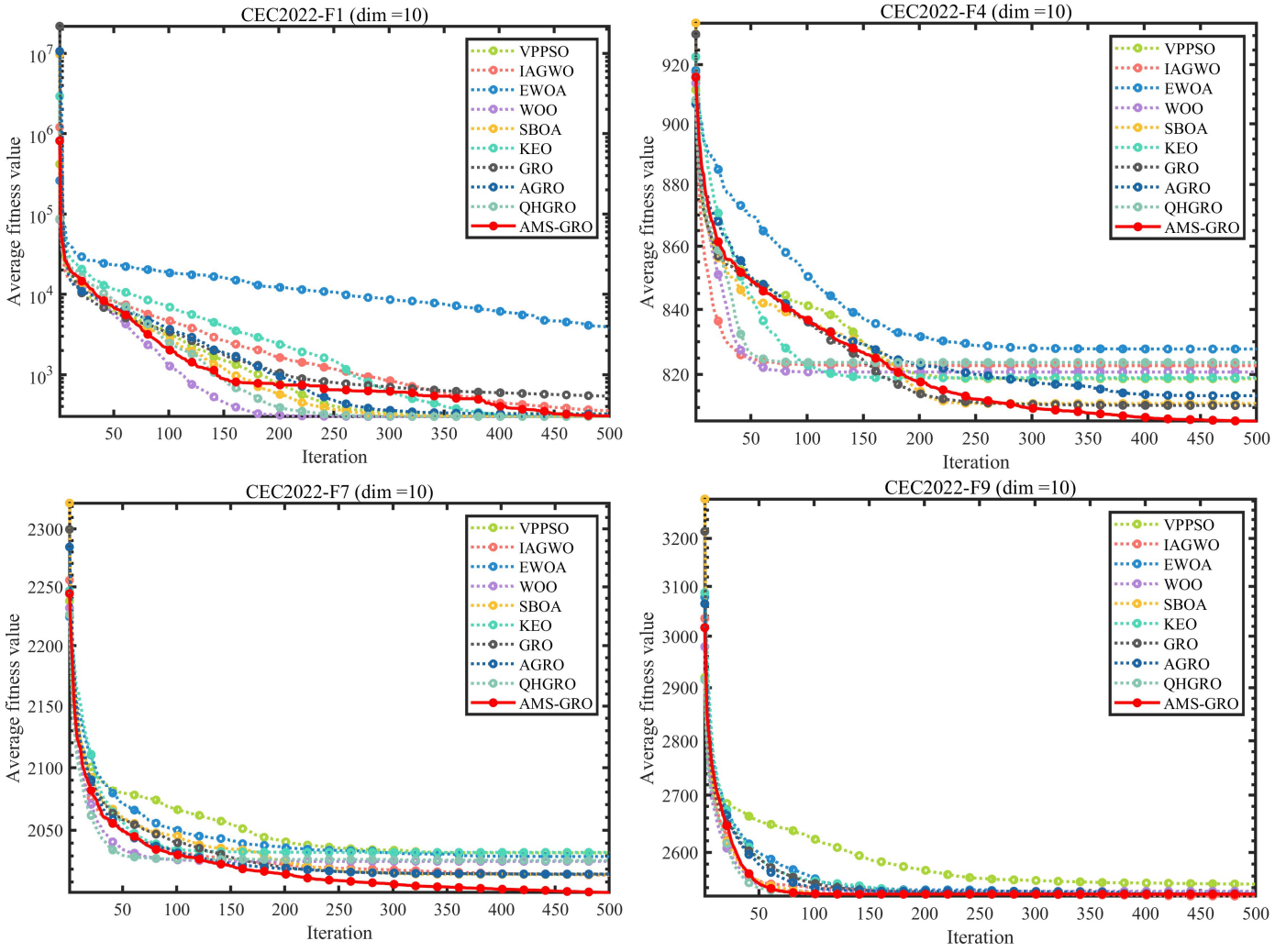


Figure 5. *Cont.*

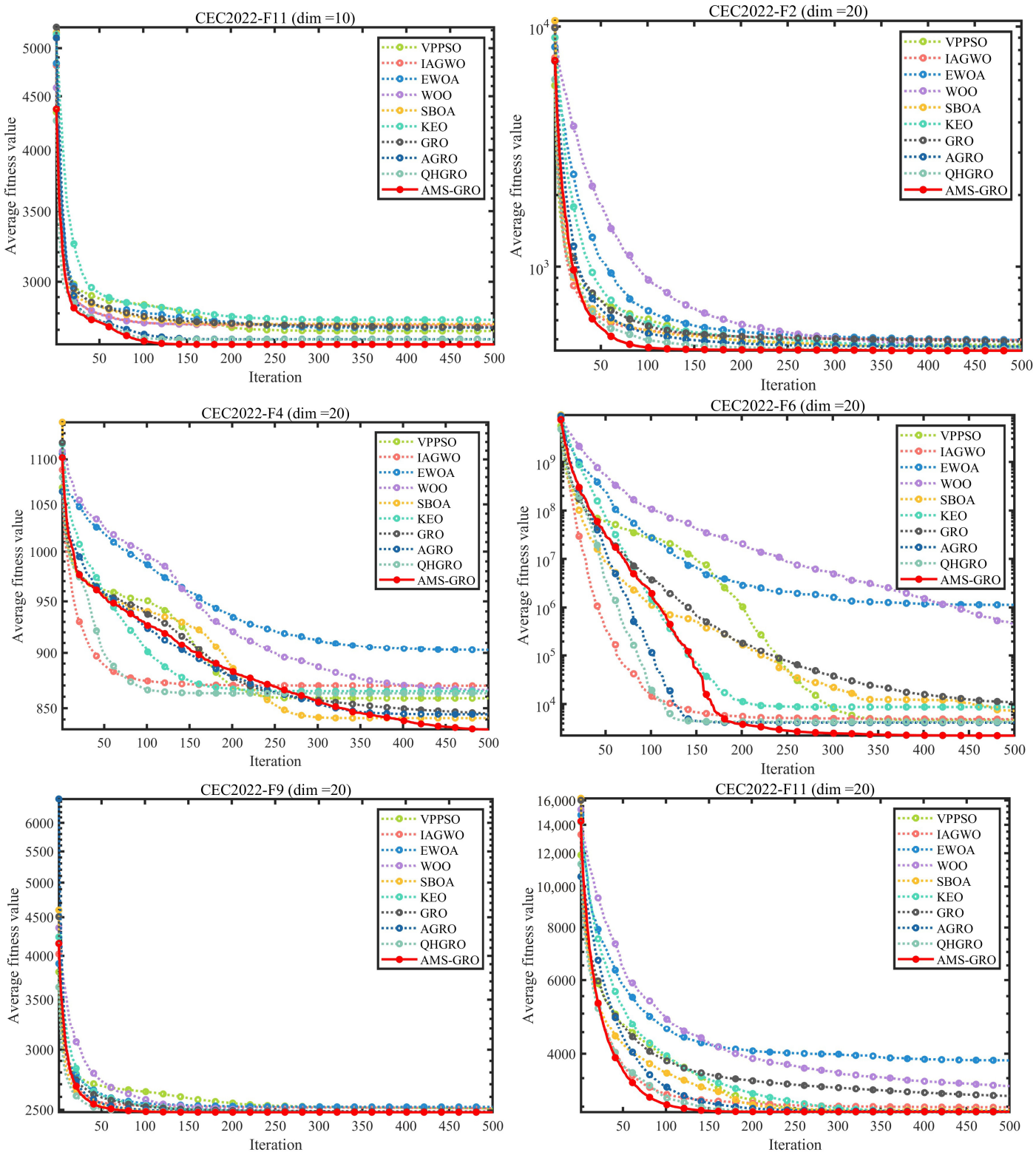


Figure 5. Convergence curves obtained on the CEC2022 benchmark functions.

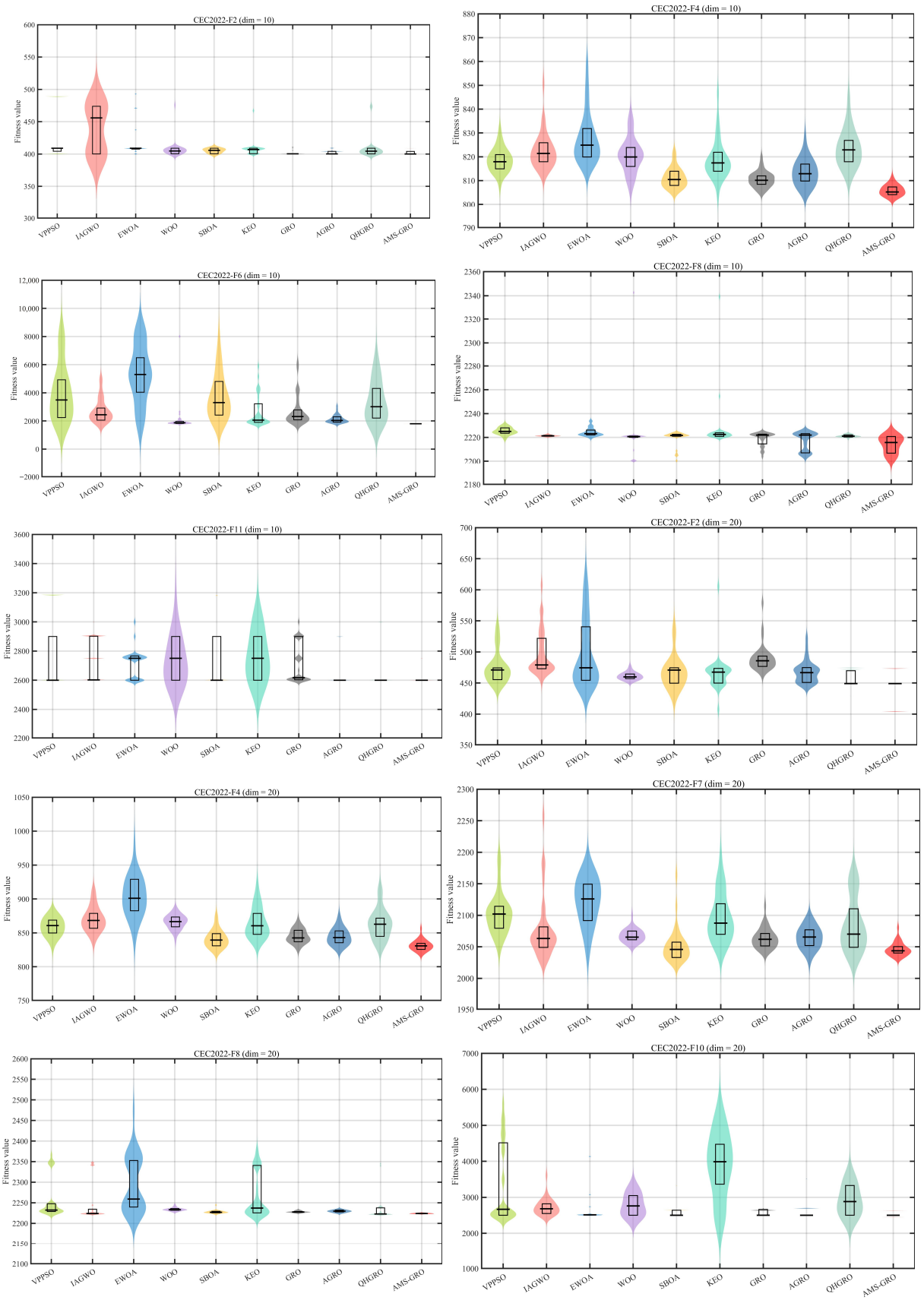


Figure 6. Boxplot comparison of methods evaluated on the CEC2022 benchmark problems.

The evolutionary trajectories depicted in Figure 5 offer a visual depiction of the search behavior exhibited by AMS-GRO when tackling the CEC2022 test problems across varying

dimensional settings. In the 10-dimensional case, the convergence curves of F4 and F7 show that AMS-GRO rapidly descends to low-fitness regions in the early stage (within the first 50 iterations). For F4, the convergence stabilizes at around 805 after 200 iterations, whereas VPPSO, IAGWO, and other algorithms fluctuate above 810 throughout the search. For F9, the convergence curve of AMS-GRO exhibits no oscillations during the entire process and converges to a final value far lower than those of other algorithms, confirming its strong capability to escape local optima in multimodal landscapes. In the 20-dimensional high-dimensional scenario, the advantages of AMS-GRO become more evident on F5 and F6. For F5, the convergence curve drops to around 900 after 300 iterations, while IAGWO, EWOA, and other algorithms remain above 1000 with severe fluctuations. For F6, the convergence curve of AMS-GRO consistently stays at the lowest level and stabilizes at approximately 2.2×10^3 after 400 iterations, which is nearly half of those achieved by AGRO and QHGRO, fully demonstrating that AMS-GRO maintains an efficient convergence rate even in high-dimensional spaces.

The boxplots in Figure 6 further verify the superior performance of AMS-GRO from a statistical distribution perspective. In the 10-dimensional case, the boxplots of F2, F4, and F6 show that AMS-GRO has the narrowest box heights, medians closest to the optimal values, and no outliers. For F6, the box of AMS-GRO is concentrated within the range of 1.8×10^3–1.81×10^3, whereas the boxes of VPPSO and EWOA span more than 3000 with frequent outliers. For F9, the box of AMS-GRO is almost coincident with the theoretical optimum, exhibiting stability far superior to that of other algorithms. In the 20-dimensional case, similar trends are observed in the boxplots of F3, F8, and F10. The box heights of AMS-GRO are significantly lower than those of the comparison algorithms. For F3, the box is concentrated around 600 with no scattered points, while the boxes of other algorithms span more than 10. For F10, the median of AMS-GRO is approximately 2510, which is far lower than those of VPPSO (3370) and IAGWO (2730), and it also exhibits the smallest interquartile range, indicating extremely high consistency across multiple runs.

Taken together, the numerical results in Tables 3 and 4, convergence profiles in Figure 5, and statistical distributions in Figure 6 demonstrate the consistent performance of AMS-GRO. Across both 10- and 20-dimensional settings, AMS-GRO achieved favorable solution accuracy, convergence speed, and run-to-run stability over the different CEC2022 function categories. This performance appears to reflect the complementary roles of the three enhancement mechanisms. The current-to-pbest/1 mutation maintains a balance between convergence pressure and population diversity, particularly as dimensionality increases. Adaptive elite-guided search strengthens local refinement, whereas success-rate-based strategy selection adjusts the search process according to the observed effectiveness of each strategy. Collectively, these findings support the robustness and generalizability of AMS-GRO across the tested CEC2022 problems and indicate its potential for broader optimization applications.

3.4. Performance Ranking of Algorithms Based on the Friedman Test

To provide a holistic quantitative assessment of the overall solution capability exhibited by AMS-GRO across diverse benchmark collections and dimensional configurations, the Friedman average rank test—a nonparametric statistical tool—is employed in this section. By calculating the mean rank (M.R.) and total rank (T.R.) for each participating algorithm, a comprehensive ranking of all competing methods is established, where lower values for both metrics signify more favorable overall performance [35]. The statistical summaries compiled in Table 5 unequivocally indicate that AMS-GRO achieves the top rank across every experimental scenario, thereby attesting to its robust generalization properties and well-rounded competitive edge.

Table 5. Comparison of Groups Based on Friedman Test Mean Ranks.

Suites	**CEC2017**		**CEC2022**			
Dimensions	**30**		**10**		**20**	
Algorithms	**M.R**	**T.R**	**M.R**	**T.R**	**M.R**	**T.R**
VPPSO	7.03	9	7.25	9	7.00	9
IAGWO	6.40	7	6.42	6	6.58	8
EWOA	9.37	10	8.42	10	9.25	10
WOO	6.97	8	4.33	4	6.33	6
SBOA	3.03	2	4.17	2	3.25	2
KEO	5.83	5	6.42	6	6.00	5
GRO	5.97	6	5.92	5	6.33	6
AGRO	4.03	3	4.25	3	4.67	4
QHGRO	5.00	4	6.50	8	4.33	3
AMS-GRO	1.37	1	1.33	1	1.25	1

For the 30-dimensional CEC2017 benchmark suite, AMS-GRO achieved the lowest mean rank (M.R. = 1.37) and ranked first overall (T.R. = 1). The next-best methods were SBOA (M.R. = 3.03, T.R. = 2) and AGRO (M.R. = 4.03, T.R. = 3). AMS-GRO also outperformed the GRO variants, including the original GRO (M.R. = 5.97, T.R. = 6) and QHGRO (M.R. = 5.00, T.R. = 4). Relative to these two variants, AMS-GRO reduced the mean rank by 77.0% and 72.6%, respectively. These outcomes provide compelling evidence that its overall optimization proficiency on high-dimensional complex test problems substantially surpasses that of all competing approaches. Turning to the CEC2022 benchmark collection, AMS-GRO similarly preserves unequivocal superiority across both reduced and elevated dimensional settings. For the 10-dimensional case, it records a mean rank of 1.33 and a total rank of 1, corresponding to mean-rank decreases of 67.9% relative to SBOA (M.R. = 4.17, T.R. = 2) and 68.7% relative to AGRO (M.R. = 4.25, T.R. = 3), which occupy the second and third positions, respectively. In the 20-dimensional high-dimensional configuration, the mean rank of AMS-GRO drops further to 1.25, while its total rank remains fixed at 1. Relative to QHGRO (M.R. = 4.33, T.R. = 3), which ranks third in this scenario, the mean-rank reduction reaches 71.1%. Notably, despite the heightened complexity inherent in higher-dimensional optimization, the solution quality delivered by AMS-GRO stays consistently robust, and the margin of superiority over other contenders becomes even more pronounced.

It is worth noting that EWOA ranks last in all scenarios (CEC2017 30-dimensional T.R. = 10, CEC2022 10-dimensional T.R. = 10, CEC2022 20-dimensional T.R. = 10), with mean ranks exceeding 8.4 in all cases, indicating an extremely large performance gap compared with AMS-GRO. Classical algorithms such as VPPSO and IAGWO generally exhibit mean ranks above 6.0, and their overall performance is far inferior to that of AMS-GRO. Moreover, although AGRO and QHGRO, which are also based on the GRO framework, show certain improvements over the original GRO, their mean ranks remain above 4.0. The consistent performance gap in favor of AMS-GRO suggests that its three enhancement mechanisms jointly improve the overall optimization capability. Specifically, current-to-pbest/1 mutation, adaptive elite-guided search, and success-rate-based adaptive strategy selection appear to provide complementary benefits rather than isolated improvements.

The stability of AMS-GRO can be attributed to the joint action of four mechanisms rather than to a single operator. First, the differential component of current-to-pbest/1 preserves directional diversity and reduces excessive dependence on the global-best solution. Second, the Gaussian perturbation in the elite-guided strategy decreases progressively, preventing large late-stage oscillations. Third, the success-rate feedback smooths the strat-

egy probabilities and retains a nonzero probability for every strategy, thereby avoiding abrupt strategy collapse. Fourth, greedy selection prevents a worse trial solution from replacing its parent. These mechanisms are reflected in several standard-deviation results. For example, on the 30-dimensional CEC2017 F5 and F6 functions, AMS-GRO obtained standard deviations of 1.2125×10^{1} and (8.2906×10^{-2}, respectively, which were the lowest among the compared methods. Nevertheless, stability is function-dependent; AMS-GRO does not obtain the smallest standard deviation on every benchmark function.

The outcomes derived from the Friedman test furnish statistical confirmation of the marked superiority exhibited by AMS-GRO. Its persistently top-tier performance, observed consistently across varying dimensional scales and diverse categories of benchmark problems, constitutes compelling testimony to both its robustness and its generalization capacity. Such statistically grounded evidence lends credible support to the deployment of AMS-GRO in real-world engineering optimization contexts—for instance, the tuning of hyperparameters within short-term power load forecasting frameworks.

3.5. Scalability Analysis Across Different Dimensions

The scalability of AMS-GRO was evaluated from both dimensional and computational perspectives. The benchmark experiments cover the 10- and 20-dimensional CEC2022 suites and the 30-dimensional CEC2017 suite. Under the same population size of 30, maximum iteration number of 500, and 30 independent runs, AMS-GRO obtained Friedman mean ranks of 1.33, 1.25, and 1.37 in the CEC2022 10-dimensional, CEC2022 20-dimensional, and CEC2017 30-dimensional settings, respectively. It ranked first in all three settings.

The comparison between the 10- and 20-dimensional CEC2022 results is particularly relevant because the same benchmark functions are evaluated at two dimensional scales. Although increasing the dimension enlarges the search space and generally increases the difficulty of locating high-quality solutions, AMS-GRO retained the first overall rank, and its mean rank changed only from 1.33 to 1.25. The 30-dimensional CEC2017 experiment provides an additional, more diverse benchmark setting, in which AMS-GRO again achieved the first overall rank with a mean rank of 1.37. These results indicate that the relative performance of AMS-GRO remains stable within the tested range of 10–30 dimensions.

This conclusion should nevertheless be interpreted within the tested dimensional range. The present experiments do not establish scalability for very high-dimensional problems, such as those involving hundreds or thousands of decision variables. Moreover, as shown by the complexity analysis, the internal computational cost increases approximately linearly with DD, apart from the $O(N\log N)$ sorting operation. Future work will therefore evaluate dimensionality-reduction and cooperative-decomposition mechanisms for large-scale optimization.

4. Short-Term Power Load Forecasting via AMS-GRO-Optimized Dilated BiGRU

This chapter presents an enhanced short-term electricity load forecasting framework that integrates the AMS-GRO with the Dilated Bidirectional Gated Recurrent Unit (Dilated BiGRU). The proposed framework aims to overcome the limitations of conventional Dilated BiGRU models, which heavily rely on manually selected hyperparameters and are therefore prone to suboptimal convergence and limited generalization performance under highly volatile load conditions.

By embedding AMS-GRO into the forecasting pipeline, key hyperparameters of the Dilated BiGRU are globally optimized in an adaptive manner. This tight coupling between intelligent optimization and deep temporal modeling enables the proposed

AMS-GRO-Dilated BiGRU model to achieve higher prediction accuracy, faster convergence, and improved robustness.

4.1. Fundamentals of the Dilated BiGRU Model

The Dilated BiGRU is an advanced recurrent neural network architecture that combines bidirectional GRU units with a dilation mechanism to efficiently capture long-term temporal dependencies. Its effectiveness in modeling complex load sequences stems from three core design components: bidirectional temporal encoding, dilated state transitions, and multi-scale feature extraction.

4.1.1. Bidirectional GRU Structure

In a standard GRU network, temporal information flows only in the forward direction, which may lead to incomplete representation of sequence dynamics. The Bidirectional GRU (BiGRU) addresses this limitation by processing the input sequence in both forward and backward directions [36].

At time step t, the hidden representation is formed by concatenating the forward hidden state $\overrightarrow{h_t}$ and the backward hidden state $\overleftarrow{h_t}$, expressed as:

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{12}$$

where h_t represents the final hidden representation obtained at time step t. This structure allows the model to simultaneously exploit historical and future contextual information, thereby improving temporal representation accuracy.

4.1.2. Dilated Temporal Connection Mechanism

Although BiGRU enhances contextual modeling, its receptive field still expands linearly with sequence length. To address this limitation, the Dilated BiGRU introduces a dilation factor d, enabling the network to establish skip connections across distant time steps.

The dilated temporal convolution operation applied to the input sequence at time step t can be expressed as:

$$y(t) = \sum_{k=1}^{K} w_k \cdot x(t - dk) \tag{13}$$

where $x(t)$ denotes the input sequence at time step t, $y(t)$ is the corresponding dilated output, w_k denotes the weight associated with the k-th kernel element, K is the kernel size. By increasing d, the effective receptive field grows exponentially without increasing network depth, allowing efficient modeling of long-range dependencies.

4.1.3. Multi-Scale Temporal Feature Modeling

Electricity load series exhibit hierarchical temporal patterns, including short-term fluctuations, daily cycles, and longer seasonal trends. The Dilated BiGRU captures these patterns by stacking multiple layers with progressively increasing dilation factors.

Lower layers focus on local variations with small dilation values, while higher layers capture broader temporal structures using larger dilation intervals. This hierarchical design enables the model to learn multi-scale temporal features in a unified framework.

As illustrated in Figure 7, the forecasting process begins with the normalization and sliding-window reconstruction of historical load and meteorological variables. The dilated temporal blocks enlarge the receptive field without requiring a proportional increase in network depth. The resulting features are subsequently processed by forward and backward GRU branches and concatenated to incorporate contextual information from both temporal directions. The attention module then emphasizes the more informative

time steps, and the weighted representation is mapped to the final load forecast through a fully connected regression layer.

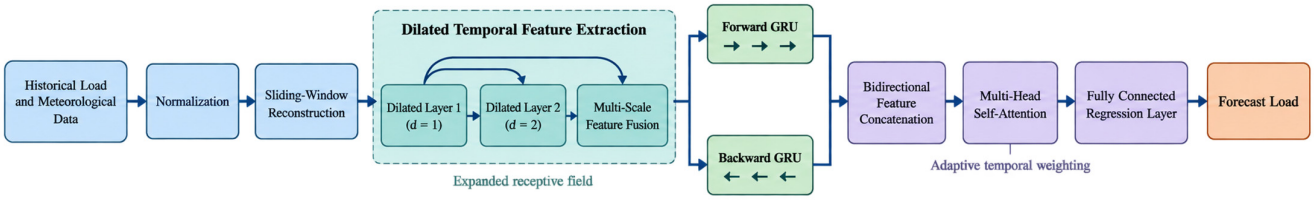


Figure 7. Flowchart of the Dilated BiGRU-Attention forecasting model.

4.2. Construction of the AMS-GRO-Dilated BiGRU Model

4.2.1. Hyperparameter Optimization Problem Formulation

The predictive performance of the Dilated BiGRU model depends strongly on several hyperparameters, including the hidden-layer size, batch size, and learning rate. These parameters interact in a nonlinear manner, making manual tuning inefficient and unreliable.

In this study, hyperparameter optimization is formulated as a global minimization problem, where the objective is to reduce prediction error on a validation dataset. AMS-GRO is employed to explore the hyperparameter space efficiently. The search ranges are summarized in Table 6.

Table 6. Search ranges of hyperparameters optimized by AMS-GRO.

Hyperparameter	Symbol	Search Ranges	Description
Learning rate	η	$[0.0001,\ 0.01]$	Initial learning rate of the Adam optimizer
Number of GRU hidden units	N_h	$[10,\ 50]$	Number of hidden units in each GRU layer
Attention key dimension	d_k	$[2,\ 50]$	Key dimension of the self-attention mechanism
L2 regularization coefficient	λ	$[1 \times 10^{-4},\ 1 \times 10^{-3}]$	Weight decay factor to prevent overfitting

From an optimization perspective, the investigated task is a bounded, continuous–discrete, derivative-free hyperparameter optimization problem. Each candidate solution is represented by a four-dimensional vector $\boldsymbol{\theta} = [\eta, N_h, d_k, \lambda]$, where η is the continuous learning rate, N_h is the integer number of GRU hidden units, d_k is the integer attention key dimension, and λ is the continuous L2 regularization coefficient. Integer-valued variables are rounded to their nearest feasible integers before model training. Because every fitness evaluation requires training and validating a deep temporal model, the objective function is computationally expensive, nonlinear, nonconvex, and unavailable in explicit analytical form. Gradient information with respect to the four hyperparameters is therefore not directly available, motivating the use of a population-based derivative-free optimizer.

4.2.2. Integration of AMS-GRO with Dilated BiGRU

The proposed AMS-GRO-Dilated BiGRU framework consists of two cooperative components:

(1) Global Hyperparameter Optimization Module: AMS-GRO employs the current-to-pbest/1 mutation strategy, adaptive elite-guided search strategy, and success-rate-based adaptive strategy selection mechanism to search for optimal hyperparameter combinations in a high-dimensional space.

(2) Load Forecasting Module: The optimized hyperparameters obtained by AMS-GRO are applied to train the Dilated BiGRU model, which performs short-term electricity load forecasting.

The fitness function guiding the optimization process is defined using the Mean Absolute Error (MAE):

$$\text{Fitness} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{14}$$

where y_i and $\hat{y}_i$ denote the actual and predicted load values, respectively, and N is the number of validation samples.

4.2.3. Optimization and Training Procedure

The optimization procedure follows these steps:

(1) Random initialization of hyperparameter candidates using AMS-GRO.
(2) Training the Dilated BiGRU model with each candidate configuration.
(3) Evaluating prediction performance using the fitness function.
(4) Updating candidate solutions through AMS-GRO iteration.
(5) Outputting the optimal hyperparameter set after reaching the maximum iteration count.

To prevent overfitting during optimization, early stopping based on validation loss is applied. A fixed data split and controlled random seeds are used to ensure reproducibility.

4.3. Short-Term Load Forecasting Using the AMS-GRO–Dilated BiGRU Framework

4.3.1. Data Sources and Preprocessing Procedures

The empirical investigation draws upon a publicly released electricity consumption record, which is accessible through the Kaggle data repository https://www.kaggle.com/code/virtual888/electricity-load-forecasting (accessed on 10 December 2025). The dataset originates from continuous power system monitoring records and spans the period from 3 January 2015 to 27 June 2020, covering a total of 2001 consecutive days with an hourly temporal resolution. Owing to its long observation horizon and uninterrupted time series structure, the dataset provides a reliable and representative basis for short-term electricity load forecasting research.

In its original form, the dataset consists of 48,048 hourly observations, each containing 17 variables. These variables include 12 meteorological features, one target variable representing national electricity demand, and several auxiliary variables related to temporal and regional information. The dataset exhibits strong temporal continuity and timeliness, which are essential for modeling complex load dynamics influenced by both weather conditions and human activity patterns.

To investigate the impact of meteorological factors on electricity demand, this study focuses on a regional subset corresponding to the toc area, covering the period from 1 May 2019, to 1 June 2020. This time span encompasses multiple seasonal cycles and includes periods of pronounced load variability, such as summer peak demand, thereby ensuring sufficient diversity of load patterns for robust model evaluation.

Four meteorological indicators are chosen as the input feature set, namely, 2-m air temperature (T2M_toc), 2-m specific humidity (QV2M_toc), total cloud cover (TQL_toc), and 2-m wind speed (W2M_toc). These variables are closely tied to building cooling and heating requirements, occupant thermal comfort levels, and household electricity usage patterns, and therefore constitute critical determinants of short-term load variability. The predictive objective is set as the national electricity demand (nat_demand), which exhibits pronounced volatility and nonlinearity over the sampling period, as reflected in the following descriptive statistics: mean = 1182.87, standard deviation = 192.07, minimum = 85.19, and maximum = 1754.88.

4.3.2. Performance Evaluation Criteria

To provide a holistic appraisal of the predictive capability of the proposed framework, four commonly employed evaluation indicators are adopted from the literature [37]:

Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{15}$$

Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{16}$$

Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{17}$$

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \overline{y}\right)^2} \tag{18}$$

where $\overline{y}$ denotes the mean of the true load values.

4.3.3. Benchmark Models and Parameter Configuration

To comprehensively evaluate the effectiveness of the proposed AMS-GRO-optimized Dilated BiGRU with attention mechanism, several widely used forecasting models were selected as benchmarks, including KELM, Random Forest (RF), LSTM, RNN, and the baseline Dilated BiGRU. All comparative models were implemented using the same training and testing datasets to ensure a fair comparison.

For the Kernel Extreme Learning Machine (KELM), a sigmoid kernel function was employed, and the regularization coefficient was set to 20. This configuration was chosen to balance nonlinear fitting capability and computational efficiency.

The Random Forest (RF) model was constructed with 200 decision trees, allowing sufficient ensemble diversity to capture nonlinear relationships between multivariate inputs and load demand while avoiding excessive overfitting.

The Long Short-Term Memory (LSTM) architecture was configured with a single hidden layer comprising 35 memory cells. Training proceeded over 1200 epochs under the Adam optimization scheme, with the learning rate initialized at 5×10^{-3}. This particular parameter assignment allows the LSTM to adequately capture long-range sequential regularities while preserving a stable training process.

For the Recurrent Neural Network (RNN), a simpler structure was adopted, comprising 10 hidden neurons and a maximum of 300 training epochs. This model serves as a baseline recurrent architecture for evaluating the advantages of gated and bidirectional structures.

The baseline Dilated BiGRU–Attention architecture comprises three principal modules: a Temporal Convolutional Network (TCN), a bidirectional gated recurrent unit network, and a self-attention mechanism. The TCN component employs two residual blocks with a filter size of 5, 64 convolutional filters, and exponentially increasing dilation factors. The BiGRU module uses 35 hidden units in each direction, followed by a self-attention layer with 2 attention heads and a key dimension of 50. The network is trained using the Adam

optimizer for 100 epochs, with an initial learning rate of 0.01 and an L_2 regularization coefficient of 1×10^{-4}.

For the AMS-GRO-optimized Dilated BiGRU-Attention model, key hyperparameters—including learning rate, number of GRU hidden units, attention key dimension, and L_2 regularization coefficient—were automatically tuned using the AMS-GRO algorithm. The search ranges were defined as [0.0001, 0.01] for the learning rate, [10, 50] for the number of GRU units, [2, 50] for the attention key dimension, and $[1 \times 10^{-4}, 1 \times 10^{-3}]$ for the regularization coefficient. The optimal parameter combination obtained through evolutionary optimization was then used to retrain the final forecasting model.

4.4. Experimental Findings and Interpretive Analysis

To validate the practical effectiveness of the AMS-GRO-Dilated BiGRU model in short-term power load forecasting, this section comprehensively analyzes its predictive performance and advantages by jointly considering the prediction error distribution, training convergence characteristics, prediction curve fitting accuracy, and multi-metric quantitative comparisons.

The error distribution over the training set is visualized in Figure 8. The residuals are predominantly confined to the interval between -50 and 50, with the overall spread exhibiting a reasonably even pattern and no discernible systematic skew. This observation suggests that the model has adequately captured the underlying regularities embedded in the load data throughout the training phase, without manifesting any persistent overestimation or underestimation tendencies. A very small proportion of error points exceed ± 100, which may be associated with extreme weather conditions or sudden electricity consumption events. However, these outliers do not significantly affect the overall prediction trend, suggesting that the model demonstrates strong fitting capability for typical load scenarios while maintaining a certain level of robustness against disturbances.

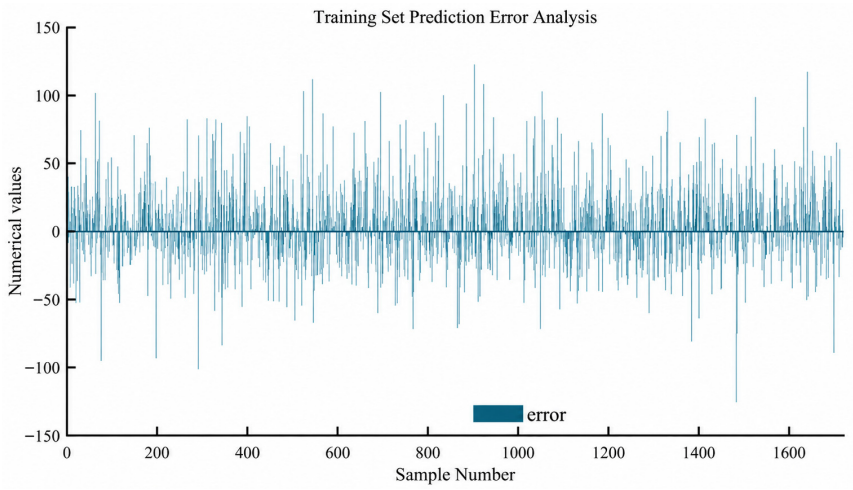


Figure 8. Distribution of Prediction Residuals on the Training Set.

Figure 9 compares the training and validation loss curves of the Dilated BiGRU model before and after optimization. For the unoptimized Dilated BiGRU, the training loss decreases slowly, and the validation loss exhibits pronounced oscillations and even an upward trend after 400 epochs, indicating a risk of overfitting. In contrast, the training loss of AMS-GRO-Dilated BiGRU rapidly drops below 0.2 within 200 epochs and then converges stably. Meanwhile, the validation loss remains consistent with the training loss throughout the training process, showing no noticeable fluctuations and eventually stabilizing at approximately 0.1. This clear difference indicates that the hyperparameter configuration optimized by AMS-GRO (e.g., learning rate and number of hidden units)

effectively balances the model's fitting ability and generalization performance, thereby avoiding both overfitting and underfitting and significantly improving training stability and convergence efficiency.

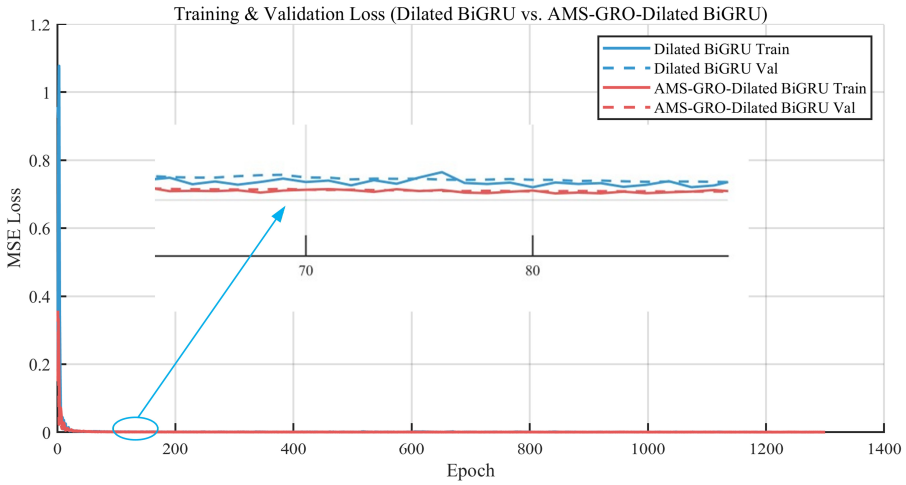


Figure 9. Training and Validation Loss Curves of Dilated BiGRU Prior to and After Hyperparameter Optimization.

Figure 10 compares the chronological actual and predicted load curves obtained using the different forecasting models. Unlike an aggregate error metric, the chronological curve retains the temporal position of each observation and therefore allows the forecasting behavior around load peaks, valleys, and abrupt transitions to be examined directly. The actual load series exhibits repeated fluctuations and several evident turning points, demonstrating the nonlinear and time-varying nature of short-term electricity demand.

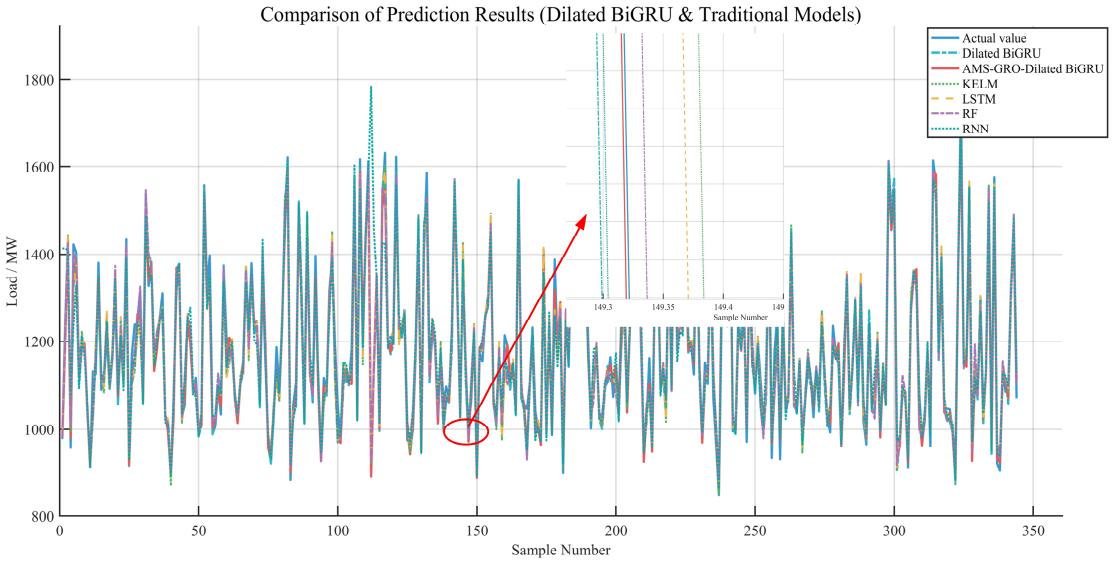


Figure 10. Forecasting Performance Comparison Across Different Models.

Among the compared methods, the forecasting curve produced by AMS-GRO–Dilated BiGRU generally follows the overall variation pattern of the actual load more closely. The proposed model reproduces most of the rising and falling trends and maintains relatively small deviations over the investigated period. The benchmark models can capture the general load tendency, but comparatively larger deviations or delayed responses can be observed around some rapidly changing intervals. These observations are consistent

with the quantitative results in Figure 10, where AMS-GRO–Dilated BiGRU obtains the lowest MAE, MAPE, and MSE and the highest R^2R^2. Among the compared methods, the forecasting curve produced by AMS-GRO–Dilated BiGRU generally follows the overall variation pattern of the actual load more closely. The proposed model reproduces most of the rising and falling trends and maintains relatively small deviations over the investigated period. The benchmark models can capture the general load tendency, but comparatively larger deviations or delayed responses can be observed around some rapidly changing intervals. These observations are consistent with the quantitative results in Figure 11, where AMS-GRO–Dilated BiGRU obtains the lowest MAE, MAPE, and MSE and the highest R^2.

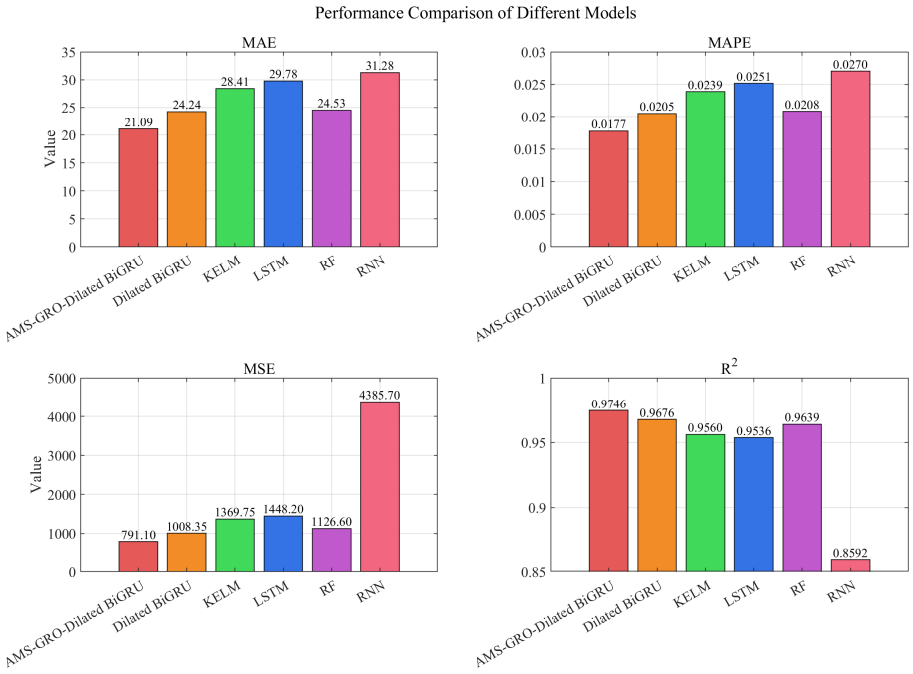


Figure 11. Visualization of prediction result metrics.

A load-duration curve is commonly constructed by arranging the load observations in descending order and is useful for describing the proportion of time during which a particular load level is equaled or exceeded. However, this rearrangement removes the chronological correspondence between the actual and predicted observations. Consequently, it cannot determine whether a peak or valley is predicted at the correct time. Because the present study focuses on short-term forecasting performance, the chronological load curves shown in Figure 10 are used as the primary graphical evidence. They preserve both the magnitude and occurrence time of the load variations and are therefore more directly related to the forecasting objective.

From an energy-management perspective, accurate short-term load forecasts can provide useful advance information for peak shaving and valley filling. When a high-load interval is predicted, energy-storage discharge, controllable generation, or demand-response resources can be scheduled in advance to reduce pressure on the power system. During a predicted low-load interval, storage charging and transferable loads can be arranged to increase valley-period utilization. Accurate identification of the transition between these operating conditions can also reduce unnecessary switching and improve scheduling reliability.

Nevertheless, the present study develops a load forecasting model rather than an energy-dispatch optimization model. Therefore, the above peak-shaving and valley-filling benefits should be regarded as potential operational applications of the forecasting re-

sults. Direct quantification of peak reduction, valley filling, operating cost, and storage degradation would require additional information concerning storage capacity, charging and discharging efficiency, electricity prices, demand-response constraints, and dispatch objectives. These aspects will be investigated in future work.

Figure 11 quantitatively compares the short-term power load forecasting performance of AMS-GRO-Dilated BiGRU with other models using four core metrics: MAE, MAPE, MSE, and R^2. The performance gaps clearly highlight the advantages of the AMS-GRO-optimized model.

In terms of MAE (Mean Absolute Error), AMS-GRO-Dilated BiGRU achieves the lowest value among all models, at 21.09. This represents a reduction of 13.0% compared with the baseline Dilated BiGRU (24.24), 25.8% compared with KELM (28.41), 29.2% compared with LSTM (29.78), and 32.6% compared with RNN (31.28). Although the reduction relative to RF (24.53) is slightly smaller (14.0%), AMS-GRO-Dilated BiGRU still demonstrates superior capability in controlling absolute prediction deviations, yielding the smallest average gap between predicted and actual load values.

For the MAPE (Mean Absolute Percentage Error) metric, AMS-GRO-Dilated BiGRU maintains the leading performance with a value of 0.0177. This corresponds to reductions of 13.7% relative to the baseline Dilated BiGRU (0.0205), 25.9% relative to KELM (0.0239), 29.5% relative to LSTM (0.0251), and 34.4% relative to RNN (0.0270). Even compared with the relatively strong RF model (0.0208), the MAPE of AMS-GRO-Dilated BiGRU is still 14.9% lower, indicating more stable relative prediction errors.

With respect to MSE (Mean Squared Error), AMS-GRO-Dilated BiGRU achieves a value of 791.10, which is substantially lower than those of all comparison models. Specifically, it is reduced by 21.5% compared with the baseline Dilated BiGRU (1008.35), 42.2% compared with KELM (1369.75), 45.4% compared with LSTM (1448.20), and 82.0% compared with RNN (4385.70). This reduction indicates that AMS-GRO-Dilated BiGRU can effectively decrease overall prediction errors and improve forecasting robustness.

In terms of R^2 (coefficient of determination), AMS-GRO-Dilated BiGRU attains a value of 0.9746, indicating a strong goodness of fit for the power load prediction task. Compared with the baseline Dilated BiGRU (0.9676), R^2 is improved by 0.70 percentage points; improvements of 1.86, 2.10, and 11.54 percentage points are observed relative to KELM (0.9560), LSTM (0.9536), and RNN (0.8592), respectively. Even compared with the relatively strong RF model (0.9639), the R^2 of AMS-GRO-Dilated BiGRU is higher by 1.07 percentage points, demonstrating its stronger capability in capturing load time-series characteristics.

By comparison, although the baseline Dilated BiGRU possesses basic time-series modeling capability, it lacks global hyperparameter optimization, resulting in inferior performance across all metrics relative to AMS-GRO-Dilated BiGRU. Traditional models such as KELM and LSTM are constrained by their algorithmic structures or optimization strategies and thus exhibit limitations in error control and goodness of fit. Overall, across all four evaluation metrics, AMS-GRO-Dilated BiGRU achieves superior performance in prediction accuracy, stability, and fitting capability, thereby validating the effectiveness of integrating hyperparameter optimization with the Dilated BiGRU model.

4.5. Comparison with Existing Forecasting Frameworks

Existing short-term load forecasting studies have improved prediction performance mainly through three technical routes: optimizer-assisted hyperparameter selection, hybrid temporal feature extraction, and attention-based feature refinement. For example, optimizer-assisted LSTM and BiGRU models use population-based algorithms to replace manual parameter selection, whereas decomposition- and convolution-based models, such as VMD–TCN–BiGRU, enhance the extraction of temporal patterns at different scales. Other

hybrid frameworks combine CNN, BiGRU, and attention mechanisms to strengthen feature representation and emphasize informative variables or temporal states.

The proposed AMS-GRO–Dilated BiGRU framework is related to these studies but differs in both the optimization and forecasting components. At the optimization level, AMS-GRO does not depend on a single fixed search operator. Instead, current-to-pbest/1 mutation, the original GRO search, and adaptive elite-guided search are coordinated through success-rate feedback. Strategies that produce successful candidate solutions receive greater selection probabilities, while the imposed probability lower bound prevents any strategy from being permanently excluded. This mechanism enables the optimizer to adjust the allocation of search resources during the optimization process.

At the forecasting level, the proposed model combines dilated temporal feature extraction, bidirectional recurrent encoding, and multi-head self-attention. The dilated structure enlarges the temporal receptive field, the bidirectional GRU captures temporal dependencies from two directions, and the attention mechanism emphasizes the more informative temporal representations. Therefore, the framework integrates hyperparameter optimization and multi-scale temporal modeling within a unified forecasting procedure.

A direct numerical comparison with results reported in different publications is not conducted because the studies employ different datasets, sampling intervals, forecasting horizons, input features, preprocessing methods, and train–test partitions. Under the common experimental protocol adopted in this study, AMS-GRO–Dilated BiGRU achieved an MAE of 21.09, MAPE of 0.0177, MSE of 791.10, and R^2 of 0.9746. Compared with the unoptimized Dilated BiGRU, the proposed framework reduced MAE, MAPE, and MSE by 13.0%, 13.7%, and 21.5%, respectively, and improved R^2 by 0.70 percentage points. Because these results were obtained using the same dataset and evaluation procedure, they provide direct evidence of the contribution of AMS-GRO-based hyperparameter optimization.

5. Conclusions and Prospects

This paper addresses key challenges in short-term power load forecasting, including the strong dependence of deep learning models on hyperparameter configurations, low optimization efficiency, and limited generalization capability. To this end, an AMS-GRO–Dilated BiGRU forecasting framework based on an Adaptive Multi-Strategy Gold Rush Optimizer is proposed. By introducing a current-to-pbest/1 mutation strategy, an adaptive elite-guided search strategy, and a success-rate–based adaptive strategy selection mechanism into the original Gold Rush Optimizer, the proposed AMS-GRO significantly enhances both global exploration and local exploitation capabilities in complex high-dimensional search spaces. Extensive benchmark function experiments demonstrate that AMS-GRO consistently outperforms multiple classical and state-of-the-art swarm intelligence algorithms in terms of optimization accuracy, convergence speed, and robustness.

In the short-term power load forecasting experiments, AMS-GRO is successfully applied to the global hyperparameter optimization of the Dilated BiGRU-Attention model. The experimental results indicate that, compared with traditional machine learning models and unoptimized deep learning models, the proposed AMS-GRO–Dilated BiGRU framework achieves superior performance across multiple evaluation metrics, including MAE, MAPE, MSE, and R^2. The model is able to more accurately capture the nonlinear characteristics and multi-scale temporal dependencies of power load series, demonstrating strong predictive accuracy and robustness under complex and highly volatile load conditions.

Although the proposed approach achieves promising results in short-term power load forecasting, there remains considerable scope for further research. Future work may be extended in several directions. First, AMS-GRO can be expanded to multi-objective optimization scenarios to simultaneously balance forecasting accuracy and model complexity.

Second, additional exogenous variables, such as holiday information and socioeconomic factors, can be incorporated to further enhance the model's adaptability to complex load patterns. Third, the proposed framework can be generalized to medium- and long-term load forecasting or other energy system prediction tasks, thereby validating its applicability and practical value in a broader range of real-world engineering applications. In addition, although the forecasting results provide useful information for identifying peak- and low-load periods, the present study does not incorporate an explicit energy-storage or demand-response scheduling model. Future work will couple the proposed forecasting framework with dispatch optimization to quantitatively evaluate its effects on peak reduction, valley filling, and operating cost.

Author Contributions: Conceptualization, X.L. and Y.Y.; methodology, X.L. and Y.Y.; software, X.L. and Y.Y.; validation, X.L. and Y.Y.; formal analysis, X.L. and Y.Y.; investigation, X.L. and Y.Y.; resources, X.L. and Y.Y.; data curation, X.L. and Y.Y.; writing—original draft preparation, X.L. and Y.Y.; writing review and editing, X.L. and Y.Y.; visualization, X.L. and Y.Y.; supervision, X.L. and Y.Y.; funding acquisition, X.L. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to express their sincere gratitude to all those who contributed to the completion of this work. The authors used ChatGPT (GPT-4o) solely for language polishing, grammar checking, and improving the readability of the text. All authors have reviewed and edited the output of the tool and take full responsibility for the content of this publication. All authors have reviewed and approved the final version of the manuscript and take full responsibility for its content.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Beloev, H.I.; Saitov, S.R.; Filimonova, A.A.; Chichirova, N.D.; Babikov, O.E.; Iliev, I.K. Short-Term Electrical Load Forecasting Based on XGBoost Model. *Energies* **2025**, *18*, 5144. [CrossRef]
2. Fan, C.; Li, G.; Xiao, L.; Yi, L.; Nie, S. Short-Term Power Load Forecasting in City Based on ISSA-BiTCN-LSTM. *Cogn. Comput.* **2025**, *17*, 39. [CrossRef]
3. Liu, X.; Wei, B.; Xue, W.; Li, X.; Li, K. An ensemble transfer learning strategy for short-term electricity load forecasting of data-sparse buildings. *Eng. Appl. Artif. Intell.* **2025**, *161*, 112166. [CrossRef]
4. Sun, J.; Ma, K.; Zhao, H. Short-term power load hybrid forecasting using GRU and SCN. *Energy Build.* **2025**, *347*, 116372. [CrossRef]
5. Wang, C.; Wang, G.; Liu, Y.; Ren, S.; Wang, J. Short-term power load forecasting based on parallel decomposition. *Adv. Eng. Inform.* **2025**, *68*, 103729. [CrossRef]
6. Yu, M.; Yang, H.; Wei, X.; Fu, S.; Xu, J.; Zhang, J. Carbon Footprint Efficiency Ratio: A Unified Indicator for Green Evaluation of Evolutionary Algorithms. *IEEE Trans. Emerg. Top. Comput. Intell.* **2026**, *10*, 2887–2900. [CrossRef]
7. Badjan, A.; Rashed, G.I.; Bahageel, A.O.M.; Gony, H.A.I.; Shaheen, H.I.; Tuaimah, F.M. Efficient grid management: Smart forecasting of short-term power load using PSO-LSTM. *Eng. Res. Express* **2024**, *6*, 035364. [CrossRef]
8. Peng, Z.; Han, H.; Ma, J. Research on a Short-Term Electric Load Forecasting Model Based on Improved BWO-Optimized Dilated BiGRU. *Sustainability* **2025**, *17*, 9746. [CrossRef]
9. Yu, M.; Yang, H.; Fu, S.; Kong, D.; Xu, X.; Zhang, J.; Xu, J. Improved Coverage and Redundancy Management in WSN Using ENMDBO: An Enhanced Metaheuristic Solution. *IEEE Trans. Netw. Serv. Manag.* **2025**, *22*, 5822–5839. [CrossRef]
10. Yu, M.; Zhang, J.; Kong, D.; Zhang, K.; Fu, S.; Jiang, F.; Xu, J. Fractional-quantum reinforcement learning differential evolution for large-scale edge computing offloading. *Knowl.-Based Syst.* **2026**, *346*, 116188. [CrossRef]
11. Yu, M.; Zhang, J.; Yu, L.; Yang, H.; Lu, H.; Wei, X.; Xu, J. Alzheimer's disease brain image segmentation using multi-feature fusion in 3D Rényi entropy model and quantum hybrid optimization. *Artif. Intell. Rev.* **2026**, *59*, 95. [CrossRef]
12. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the ICNN'95-International Conference on Neural Networks*; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948.

13. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
14. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
15. Fu, Y.; Liu, D.; Chen, J.; He, L. Secretary bird optimization algorithm: A new metaheuristic for solving global optimization problems. *Artif. Intell. Rev.* **2024**, *57*, 123. [CrossRef]
16. Yu, M.; Yang, H.; Zhang, J.; Ouyang, K.; Fu, S.; Tan, P.; Jiang, F.; Xu, J. Bounty hunter optimizer: A novel metaheuristic with an application to multi-UAV mobile edge computing and path planning. *Knowl.-Based Syst.* **2026**, *341*, 115836. [CrossRef]
17. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
18. Wang, R.-B.; Hu, R.-B.; Geng, F.-D.; Xu, L.; Chu, S.-C.; Pan, J.-S.; Meng, Z.-Y.; Mirjalili, S. The Animated Oat Optimization Algorithm: A nature-inspired metaheuristic for engineering optimization and a case study on Wireless Sensor Networks. *Knowl.-Based Syst.* **2025**, *318*, 113589. [CrossRef]
19. Zhu, Y.; Zhang, M.; Huang, Q.; Wu, X.; Wan, L.; Huang, J. Secretary bird optimization algorithm based on quantum computing and multiple strategies improvement for KELM diabetes classification. *Sci. Rep.* **2025**, *15*, 3774. [CrossRef] [PubMed]
20. Wang, J.; Qu, Z.; Zang, H.; Bai, K.; Tian, L.; Zhang, R. VMD-BiGRU-ATT architecture for aviation failure events prediction based on the grey wolf optimization algorithm. *Reliab. Eng. Syst. Saf.* **2025**, *267*, 111934. [CrossRef]
21. Luo, H.; Chen, J.; Sun, Z.; Zhang, Y.; Zhang, L. Improved Marine Predators Algorithm Optimized BiGRU for Strip Exit Thickness Prediction. *IEEE Access* **2024**, *12*, 56719–56729. [CrossRef]
22. Zolfi, K. Gold rush optimizer: A new population-based metaheuristic algorithm. *Oper. Res. Decis.* **2023**, *33*, 113–150. [CrossRef]
23. Dhandapani, S.; Subburam, R.; Cyriloose, P.D.C.; Balan, S.K. Transfer Learning with Adam Gold Rush Optimization for Endometrial Disease Classification Using Histopathological Image. *Microsc. Res. Tech.* **2025**, *88*, 3063–3082. [CrossRef] [PubMed]
24. Zhang, Y.; Adegboye, O.R.; Feda, A.K.; Agyekum, E.B.; Kumar, P. Dynamic gold rush optimizer: Fusing worker adaptation and salp navigation mechanism for enhanced search. *Sci. Rep.* **2025**, *15*, 15779. [CrossRef] [PubMed]
25. Tanabe, R.; Fukunaga, A.S. Improving the search performance of SHADE using linear population size reduction. In *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC)*; IEEE: New York, NY, USA, 2014; pp. 1658–1665.
26. Awad, N.H.; Ali, M.Z.; Suganthan, P.N. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*; IEEE: New York, NY, USA, 2017; pp. 372–379.
27. Shami, T.M.; Mirjalili, S.; Al-Eryani, Y.; Daoudi, K.; Izadi, S.; Abualigah, L. Velocity pausing particle swarm optimization: A novel variant for global optimization. *Neural Comput. Appl.* **2023**, *35*, 9193–9223. [CrossRef]
28. Yu, M.; Xu, J.; Liang, W.; Qiu, Y.; Bao, S.; Tang, L. Improved multi-strategy adaptive Grey Wolf Optimization for practical engineering applications and high-dimensional problem solving. *Artif. Intell. Rev.* **2024**, *57*, 277. [CrossRef]
29. Nadimi-Shahraki, M.H.; Zamani, H.; Mirjalili, S. Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study. *Comput. Biol. Med.* **2022**, *148*, 105858. [CrossRef] [PubMed]
30. Peng, Y.; Gu, S.; Liang, Y.; Ouyang, K.; Li, Y.; Wang, K.; Wu, G.; Fan, C. Wave Optics Optimizer: A novel meta-heuristic algorithm for engineering optimization. *Commun. Nonlinear Sci. Numer. Simul.* **2025**, *152*, 109337. [CrossRef]
31. Almutairi, S.Z.; Shaheen, A.M. A novel kangaroo escape optimizer for parameter estimation of solar photovoltaic cells/modules via one, two and three-diode equivalent circuit modeling. *Sci. Rep.* **2025**, *15*, 32669. [CrossRef] [PubMed]
32. Lyu, L.; Kong, G.; Yang, F.; Li, L.; He, J. Augmented Gold Rush Optimizer is used for engineering optimization design problems and UAV path planning. *IEEE Access* **2024**, *12*, 134304–134339. [CrossRef]
33. Yu, M.; Lu, H.; Wang, D.; Du, J.; Kong, D.; Xu, X.; Xu, J. Quantum-enhanced gold rush Optimizer for multi-threshold segmentation of lupus nephritis pathological images. *Displays* **2026**, *92*, 103334. [CrossRef]
34. Chen, Y.; Sun, R.; Zheng, J.; Shao, Y.; Zhou, H. MEIAO: A Multi-Strategy Enhanced Information Acquisition Optimizer for Global Optimization and UAV Path Planning. *Biomimetics* **2025**, *10*, 765. [CrossRef] [PubMed]
35. Yang, Y.; Sun, L.; Zhang, N. An MTBWO Algorithm Based on BiGRU Model. *Electronics* **2024**, *13*, 1195. [CrossRef]
36. Zou, Z.; Wang, J.; E, N.; Zhang, C.; Wang, Z.; Jiang, E. Short-Term Power Load Forecasting: An Integrated Approach Utilizing Variational Mode Decomposition and TCN–BiGRU. *Energies* **2023**, *6*, 6625. [CrossRef]
37. Dai, L.; Wang, H. An Improved WOA (Whale Optimization Algorithm)-Based CNN-BIGRU-CBAM Model and Its Application to Short-Term Power Load Forecasting. *Energies* **2024**, *17*, 2559. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.