EnsembleFL: A Dynamic Multi-Objective Evolutionary Algorithm based on an Ensemble Feed-forward Prediction

Quanheng Zheng

Abstract—**Numerous decision-making challenges across industrial domains can be formulated as Dynamic Multi-Objective Optimization Problems (DMOPs). As an effective computational paradigm for DMOPs, Dynamic Multi-Objective Evolutionary Algorithms (DMOEAs) have attracted significant attention from both academia and industry. To address diverse environmental change patterns, this study proposes EnsembleFL, a novel DMOEA incorporating an ensemble feed-forward prediction mechanism that captures heterogeneous movement patterns of optimal solutions and accurately predicts them in new environments. Experimental evaluations on the CEC'2018 benchmark suite demonstrate EnsembleFL's superior performance compared to five state-of-the-art DMOEAs. Under severe environmental changes, EnsembleFL achieves the best mean Modified Inverted Generational Distance (MIGD) and Modified Hypervolume Difference (MHVD) values on 10 and 13 DMOPs, respectively. In scenarios with mild environmental changes, it attains optimal mean MIGD and MHVD metrics on 5 and 12 DMOPs, respectively. These results validate EnsembleFL's robustness in handling both abrupt and gradual environmental transitions, establishing it as a competitive solution for real-world dynamic optimization challenges.**

Index Terms—**Dynamic Multi-Objective Optimization, Dynamic Multiple Objective Evolutionary Algorithm, Evolutionary Algorithm, Environment Prediction, Ensemble, Feed-forward.**

I. Introduction

DIFFERENT industries face a wide array of decision-making challenges. Examples include self-driving cars [1], drone or robot path planning [2], [3], [4], [5], rail transit train schedules [6], and industrial automation [7]. These decision-making issues often involve numerous conflicting optimization objectives. Moreover, the environment may change at any point during the decision-making process, thereby influencing the optimization objectives of the decisions. As a result, Dynamic Multi-objective Optimization Problems (DMOPs) have garnered significant attention in both industry and academia.

A DMOP is an optimisation problem with multiple objectives, and its parameters and objectives may change over time. In certain situations, DMOPs have a number of constraints, some of which may be dynamic. A general DMOP can be formalised as the Eq. (1). $F(X;t)$ include $n(t)$ optimization objective functions, where these functions and their number may vary over time t. $f_i(X;t)$ is the ith objective function. $X = (x_1, ..., x_{m(t)})$ represents the vector consisting of decision variables. $\Omega(t)$ is the decision space that constrains the range of values for each decision variable, i.e., the search space of DMOP solving algorithms. $G(X;t)$ and $H(X;t)$ are the inequality and equality constraints, respectively. t represents the time that the DMOP (environment) changes. T is the number of that DMOP changes. If a DMOP is a long-term optimisation problem or its number of changes is unknown, T can be set as infinity ($+\infty$).

$$\begin{aligned}
\text{Minimizing} \quad & F(X;t) = (f_1(X;t), ..., f_{n(t)}(X;t)), && (1)\\
\text{subject to,} \quad & G(X;t) \leq 0, && (2)\\
& H(X;t) = 0, && (3)\\
& X \in \Omega(t), && (4)\\
& t = t_0, t_1, ..., t_T. && (5)
\end{aligned}$$

Dynamic Multi-Objective Evolutionary Algorithms (DMOEAs) are among the most effective approaches for solving DMOPs. However, tackling the unpredictability of dynamically changing environments in real-world scenarios poses significant challenges for enhancing the performance of DMOEAs. Specifically, DMOEAs need to: (1) accurately evaluate whether individuals in the current population maintain high quality in new environments, so as to determine which solutions should be preserved. (2) analyse environmental change patterns to generate adaptive solutions that enhance population diversity and steer evolution towards new Pareto-optimal solution sets.

Therefore, some current research endeavors are centered on predicting environmental changes. These studies construct individuals that are adapted to the new environment to guide the population toward evolving in a new optimal direction. Certain studies employ the movement direction and step size of the Pareto front centroids or the flow patterns between two consecutive recent environments. They generate new individuals based on the most recent Pareto front centroids or flow patterns. This type of method assumes that DMOPs consist of a series of stable Multi-objective Optimization Problems (MOPs). It posits that in consecutive environments, the Pareto fronts or Pareto-optimal solution sets of the MOPs are mostly similar. As a result, these methods have a relatively limited scope of application.

To bridge these gaps, this paper introduces a novel DMOEA, EnsembleFL, that integrates ensemble learning principles with multiple prediction mechanisms. By fusing two distinct feed-forward predictors based on key-points or

Manuscript received February 25, 2025; revised May 17, 2025.

Q. Zheng is a postgraduate student of the School of Computer Science & Technology, Xi'an University of Posts and Telecommunications, Xi'an, 710121 China. (corresponding author; e-mail: 18839555678@163.com)

reference points through a framework inspired by Bagging, EnsembleFL adapts to heterogeneous environmental changes and generates high-quality initial populations for new environments encountered in DMOPs. Evaluations conducted on the CEC 2018 benchmark demonstrate its superior performance. Under severe environmental changes, EnsembleFL achieves the optimal mean Modified Inverted Generational Distance (MIGD) and Modified Hypervolume Difference (MHVD) values on 10 and 13 DMOPs, respectively. Under mild environmental changes, it attains the optimal MIGD and MHVD values on 5 and 12 DMOPs, outperforming five state-of-the-art DMOEAs. This work enhances the robustness of DMOEAs in dynamic environments through the systematic integration of predictive mechanisms, providing practical solutions for real-world DMOPs characterized by unpredictable or nonlinear dynamics.

The structure of this paper is organized as follows. Section II introduces the fundamental concepts of DMOPs. Section III elaborates on the proposed ensemble prediction framework. Section IV presents the results of benchmark evaluations. Section V discusses the innovations in comparison to existing works. Section VI concludes the paper and outlines future research directions.

II. Preliminary Knowledge

Definition 1: **Pareto dominating**. In time t for a DMOP, given any two available solution, $X_1, X_2 \in \Omega(t)$, X_1 dominates X_2 ($X_1 \prec X_2$) if and only if X_1 has not greater values than X_2 for all of the objective functions, and X_1 has lower value than X_2 for at least one objective function. This can be formulated as Eq. (6). If an available solution is not dominated by any other solution, then we can call it a non-dominated solution.

$$\begin{aligned} X_1 \prec X_2 := \quad & \forall_i (f_i(X_1;t) \leq f_i(X_2;t)) \\ & \wedge \exists_i (f_i(X_1;t) < f_i(X_2;t)) \end{aligned} \tag{6}$$

Definition 2: **Pareto optimal Set (*PS*)**. *PS* is the set that consists of all available solutions that are not dominated by any solution for a DMOP at every time. This can be formulated as Eq. (7).

$$PS(t) := \{X | X \in \Omega(t) \wedge \neg\exists_{X' \in \Omega(t)} (X' \prec X)\} \tag{7}$$

Definition 3: **Pareto Frontier (*PF*)**. *PF* is the mapping of *PS* in the objective space, which can be represented as Eq. (8).

$$PF(t) := \{F(X;t) | X \in PS(t)\} \tag{8}$$

III. DMOEA Based on the Ensemble Feed-forward Prediction

In real-life scenarios, DMOP environments change in a variety of ways. Therefore, prediction mechanisms that only capture a single change pattern are difficult to adapt to environmental changes, resulting in limited performance. For this reason, this section proposes to integrate multiple prediction mechanisms to capture multiple types of environmental change patterns simultaneously to improve the accuracy of *PS* movement prediction. In this paper, we design the ensemble method to fuse two feed-forward prediction methods to capture both *PS* linear moving patterns and accelerated moving patterns. In future work, we will explore the integration of more types of prediction methods to further improve the accuracy of prediction.

Fig. 1 illustrates the technical framework of the dynamic multi-objective evolutionary algorithm with the ensemble prediction mechanism (named as EnsembleFL) designed in this paper. The core concept involves utilizing two feed-forward prediction mechanisms (forward-looking Velocity, FLV and forward-looking Acceleration, FLA) to generate two sub-populations (sub-population 1 and 2) based on the *PS* from the most recent three environments. Simultaneously, the population update strategy from DNSGA-II-B is employed to perform mutation operations on a certain proportion of randomly selected individuals from the current population, thereby creating another sub-population (sub-population 0) to enhance population diversity. These three generated sub-populations are then randomly used to replace individuals in the current population, forming a new population that serves as the initial population for the subsequent environment. For details of the DNSGA-II-B population update strategy, please refer to the cited literature [8]. The following contents will provide detailed explanations of the FLV and FLA prediction methods.

At first, to effectively capture the evolutionary patterns of *PF*, the FLV and FLA predictions are based on multiple key points rather than solely focusing on the *PF* centroid. The key points considered in this work include the *PF* centroid and uniformly sampled statistical points on the *PF*. Specifically, the *PS* is first sorted according to each dimension's values, followed by uniform sampling of the 0–100% percentile points (in this work, 11 sampling points are selected, i.e., the $10 * i\%$ percentile points, where i ranges from 0 to 10). Key points across different environments are aligned one-to-one based on their percentile positions. Subsequently, as illustrated in Figure 1, the movement of each key point (including the centroid) is predicted using the proposed technical framework. The integrated prediction methodology introduced here is also applicable to other critical points, such as knee points.

FLV (forward-looking Velocity) is one of the most classic and widely used prediction mechanisms in DMOEAs. It assumes that the *PS* undergoes constant-velocity motion and adopts the positional difference between the two most recent environments as the movement velocity for the new environment. The specific prediction method involves, for each key point, first calculating its movement velocity based on its positions in the two most recent environments. Then, using the current position and the calculated velocity, the position in the new environment is predicted. The detailed computational formulas are given in Eq. (9) and (10), where $X^{(t)}$ and $V^{(t)}$ represents the position and the velocity of a key point at time (environment) t, and $\widehat{X}^{(t+1)}$ denotes the predicted position of the key point at time t. Finally, FLV introduces Gaussian noise to each dimension to generate a specified number of individuals, forming a sub-population.

$$V^{(t)} = X^{(t)} - X^{(t-1)} \tag{9}$$

$$\widehat{X}^{(t+1)} = X^{(t)} + V^{(t)} \tag{10}$$

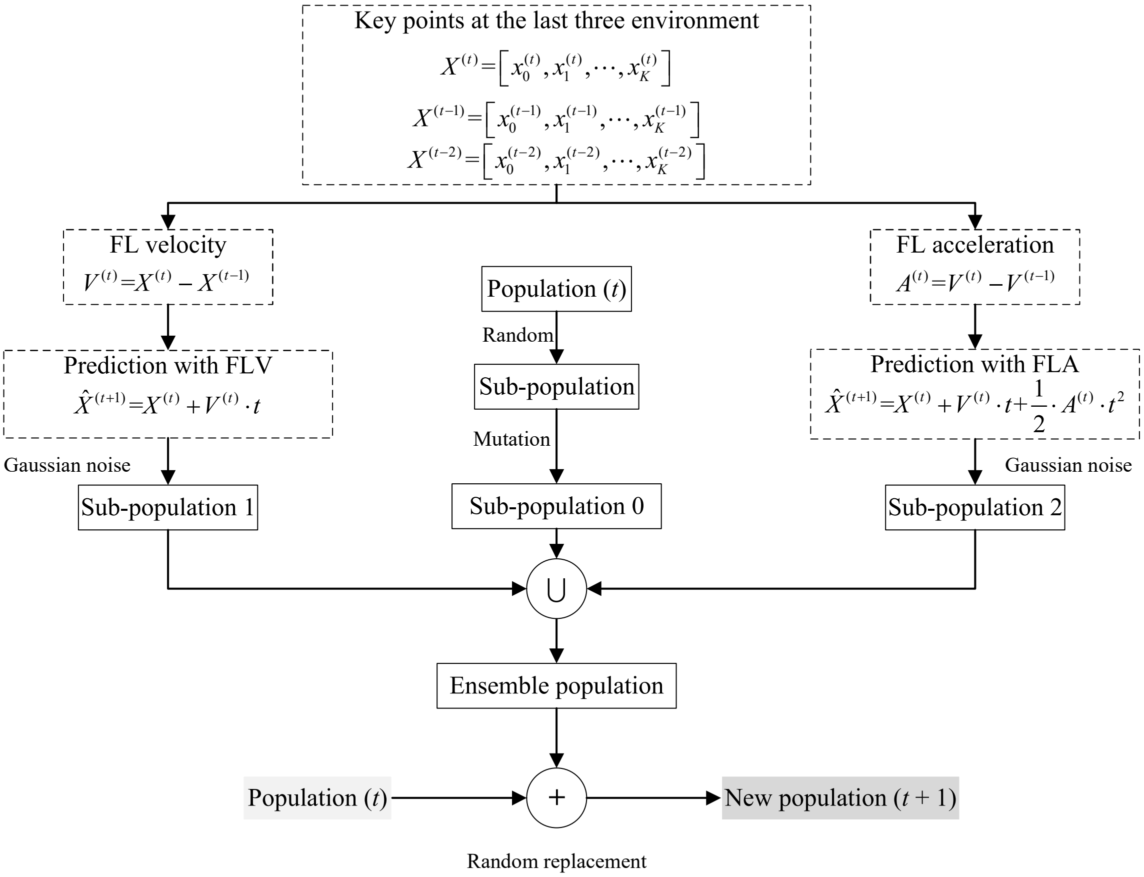

Key points at the last three environment
$X^{(t)}=\left[x_0^{(t)},x_1^{(t)},\cdots,x_K^{(t)}\right]$
$X^{(t-1)}=\left[x_0^{(t-1)},x_1^{(t-1)},\cdots,x_K^{(t-1)}\right]$
$X^{(t-2)}=\left[x_0^{(t-2)},x_1^{(t-2)},\cdots,x_K^{(t-2)}\right]$
FL velocity
$V^{(t)}=X^{(t)}-X^{(t-1)}$
Population (t)
FL acceleration
$A^{(t)}=V^{(t)}-V^{(t-1)}$
Random
Prediction with FLV
$\hat{X}^{(t+1)}=X^{(t)}+V^{(t)}\cdot t$
Sub-population
Prediction with FLA
$\hat{X}^{(t+1)}=X^{(t)}+V^{(t)}\cdot t+\frac{1}{2}\cdot A^{(t)}\cdot t^2$
Mutation
Gaussian noise
Gaussian noise
Sub-population 1
Sub-population 0
Sub-population 2
∪
Ensemble population
Population (t)
+
New population (t + 1)
Random replacement

Fig. 1. Technical framework of the ensemble feed-forward prediction mechanism. (FL: Forward-Looking; FLV: Forward-Looking Velocity; FLA: Forward-Looking Acceleration)

The FLV mechanism assumes that each key point moves at a constant velocity, leading to suboptimal performance in environments with variable-speed dynamics. To address this limitation, this paper proposes the FLA (forward-looking Acceleration) mechanism, which hypothesizes that the velocity of each key point changes over time. The core idea is first calculating the acceleration of a key point based on its velocities in the two most recent environments (derived from Eq. (9)). Then, the latest position of the key point is predicted using the computed acceleration and the most recent velocity. Detailed computations are provided in Eq. (11) and (12), where $A^{(t)}$represents the acceleration of the key point at time t. Similar to FLV, FLA incorporates Gaussian noise into the predicted positions of key points to generate a sub-population of the specified size.

$$A^{(t)} = V^{(t)} - V^{(t-1)} \tag{11}$$

$$\widehat{X}^{(t+1)} = X^{(t)} + V^{(t)} + \frac{1}{2}A^{(t)} \tag{12}$$

Based on the aforementioned ensemble feed-forward prediction mechanism, this paper designs a new DMOEA, named EnsembleFL, as outlined in Algorithm 1. First, EnsembleFL initializes a population and the current environment (Lines 1–2) and evaluates the objective function values of all individuals in the population (Line 3). It then employs the Fast Non-Dominated Sorting (FNDS) algorithm to rank the population (Line 4) and identifies the *PF* and *PS* (Line 5). After initializing each key point and its movement velocity (set to zero) (Lines 6–7), EnsembleFL enters the iterative population evolution phase (Lines 8–25) until termination criteria are met, returning the searched *PS* and *PF* (Line 26).

During the iterative evolution process, EnsembleFL operates as follows:

- When no environmental change is detected, it evolves the population using a multi-objective evolutionary algorithm (NSGA-II is adopted in this work) (Line 23).
- When an environmental change occurs, EnsembleFL updates the current population using the ensemble feed-forward prediction mechanism (Lines 10–18) and correspondingly updates the PS and PF based on FNDS (Lines 19–21). The detailed steps are outlined below:

1) Archive key points and velocities from the environment before the change (Lines 10 and 12).
2) Identify the latest positions of key points from the PS of the most recent environment (Line 11).
3) Compute movement velocities for each key point using their archived positions and the latest positions (Line 13).
4) Calculate accelerations for each key point based on their archived velocities and the newly computed velocities (Line 14).
5) Predict post-change positions of key points using FLV and FLA mechanisms (Lines 15–16).
6) Generate sub-populations of specified size (20% of the population size in this work) via mutation operators

Algorithm 1 EnsembleFL

```
Input: $F$; $\Omega$
Output: $PS$ and $PF$;
 1: Randomly initialize the population, $pop$;
 2: $t \leftarrow t_0$; //initialize the time/environment
 3: $FIT \leftarrow F(pop, t)$; //evaluate the objective function values
 4: $L \leftarrow FNDS(FIT)$; //sort by FNDS
 5: $PF \leftarrow F[L[0]]; PS \leftarrow pop[L[0]]$; //get the current $PS$ and $PF$
 6: $V \leftarrow 0$; //initializing the velocities
 7: $keys \leftarrow$ key points of $PS$ ($X^{(0)}$); //initialize the key points
 8: while not reach termination condition do
 9:     if environment is changed then
10:         $keysLast \leftarrow keys$; //archive the last key points
11:         $keys \leftarrow$ key points of $PS$ ($X^{(t)}$); //update the key points
12:         $VLast \leftarrow V$; //record the last velocities
13:         $V \leftarrow keys - keysLast$; //update the velocities
14:         $A \leftarrow V - VLast$; //calculate the accelerations
15:         $X\_FLV \leftarrow noise(keys + V)$; //build a sub-population with FLV
16:         $X\_FLA \leftarrow noise(keys + V + A/2)$; //build a sub-population by FLA
17:         $X\_B \leftarrow mutation(randSelect(pop, 20\%))$; //generate a sub-population with the idea of DNSGA-II-B
18:         $pop \leftarrow update\_pop(pop, X\_FLV \cup X\_FLA \cup X\_B)$; //update the population with random replacements
19:         $FIT \leftarrow F(pop, t)$; //re-evaluate the objective function values
20:         $L \leftarrow FNDS(FIT)$; //sort by FNDS
21:         $PF \leftarrow F[L[0]]; PS \leftarrow pop[L[0]]$; //update the current $PS$ and $PF$
22:     end if
23:     evolve pop and update $PS$ and $PF$ by NSGA-II;
24:     $t \leftarrow t + tic$; // going to the next tick
25: end while
26: return $PS$ and $PF$;
```

and random selection from the current population (Line 17).

7) Update the current population by randomly replacing individuals with the three above generated sub-populations to form a new population (Line 18).
8) Evaluate objective function values of the updated population in the new environment (Line 19).
9) Re-rank the population using FNDS to obtain the updated *PS* and *PF* (Lines 20–21).

This framework ensures adaptability to dynamic environments while maintaining population diversity and prediction accuracy through integrated velocity-acceleration modeling and stochastic perturbations.

IV. Performance Evaluation

This paper evaluates the performance of the proposed algorithm using the CEC 2018 benchmark suite [9], which comprises 14 DMOPs (DF1–DF14) with varied changing pattern of PS and PF. The details of these 14 DMOPs are shown in Table IV-IV.

The selected baseline algorithms include five recently proposed prediction-based DMOEAs: Forward-Looking (FL, i.e., FLV) [10], FLA, Linear Regression (LR), Support Vector Regression (SVR) [11], [12], and eXtreme Gradient Boosting (XGB) [13]. To ensure an objective comparison of prediction performance, all DMOEAs adopt NSGA-II as their evolutionary strategy.

The parameter settings for the test problems follow the CEC 2018 specifications [9], while experimental environment parameters are configured as shown in Table IV, covering two scenarios: severe environmental changes ($n_t = 10$) and mild environmental changes ($n_t = 30$), where n_t represents the severity of environmental changes of DMOPs and a less value represents a more severity.

The algorithms are implemented in Python, with arithmetic and matrix operations handled by the Numpy library (version 1.26.4). The experiments are conducted on a Windows 11 Home system with the following hardware: 14th Gen Intel Core i7-14700(F) processor, 16GB DDR5-5600 RAM and 1TB PCIe 4.0 SSD. All code is executed under Python 3.11.7, ensuring consistency and reproducibility across trials.

A primary criterion for evaluating the performance of a DMOEA is to quantify the discrepancy between the obtained PF (PF^{act}) and the theoretical PF. This paper employs the two most widely adopted metrics, Modified Inverted Generational Distance (MIGD) [14] and Modified Hypervolume Difference (MHVD) [15], to measure the performance of DMOEAs.

The Inverted Generational Distance (IGD) is widely used to evaluate the accuracy of multi-objective optimization algorithms. Its fundamental idea is to calculate the mean distance between each solution obtained by the algorithm and the theoretical PF. The specific calculation formula is shown in Eq. (13), where $d(f, PF(t))$ represents the shortest distance from a solved solution with objective function values of f to the theoretical PF. It can be easily observed that IGD does not take into account the temporal dynamics of DMOPs or DMOEAs. Therefore, Modified IGD (MIGD) addresses this limitation by averaging the IGD values over each change period, thereby assessing the overall performance of DMOP solving algorithms throughout the entire time span, as shown in Eq. (14).

$$IGD(PF^{act}, PS(t)) = \sum_{f \in PF^{act}} \frac{d(f, PS(t))}{|PF^{act}|} \qquad (13)$$

$$MIGD = \frac{1}{T} \sum_{\tau \in [t_0, t_T]} IGD(PF^{act}(\tau), PS(\tau)) \qquad (14)$$

A smaller MIGD value indicates that the solutions obtained by the algorithm are closer to the theoretical PF, reflecting better convergence performance. However, MIGD fails to account for regions of PF that are not covered by the obtained solution set, thereby lacking sensitivity to diversity assessment. To address this limitation, MHVD serves as a complementary metric. It quantifies the hypervolume between the obtained solution set and the theoretical PF, simultaneously evaluating both diversity (coverage of the PF) and accuracy (proximity to the PF) of the algorithm.

The Hypervolume (HV) metric measures the volume of the objective space region bounded by a reference point and the obtained non-dominated solution set. Its calculation is defined by Eq. 15, where $\mathcal{L}$ denotes the Lebesgue measure (used to quantify volume), $PF^{act}(t)$ is the PF solved by a DMOEA, and V_f is the hypervolume between

TABLE I
ILLUSTRATIONS OF PS AND PF OF DF1-DF5 IN THE CEC 2018 BENCHMARK SUITE [9]

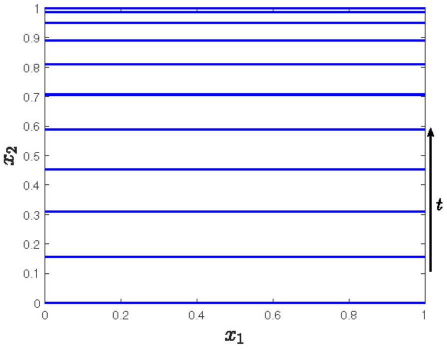
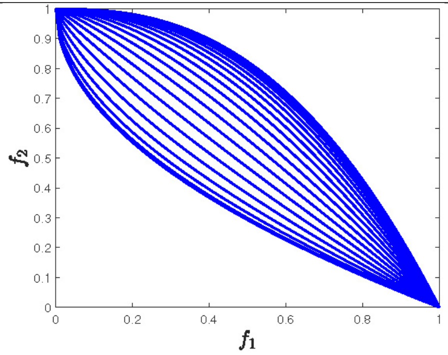
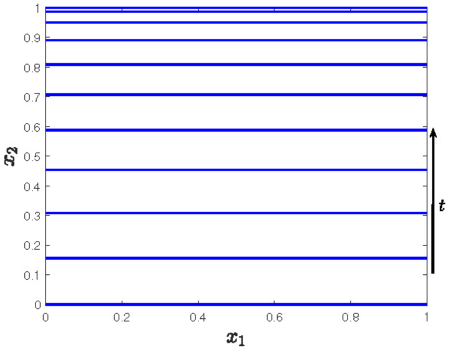
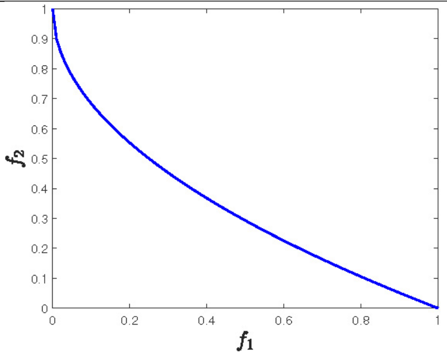
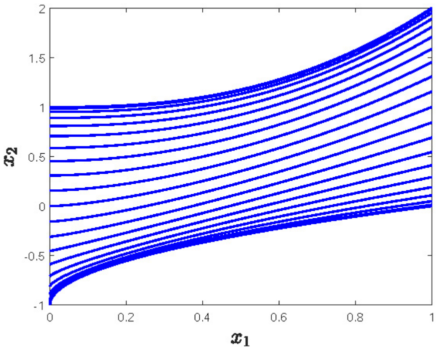
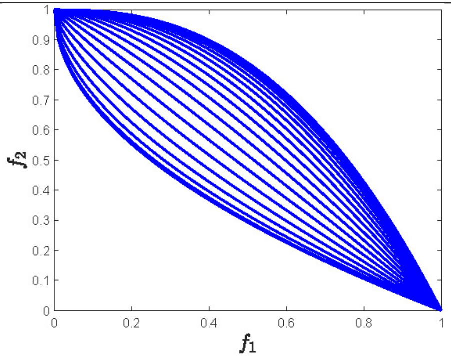
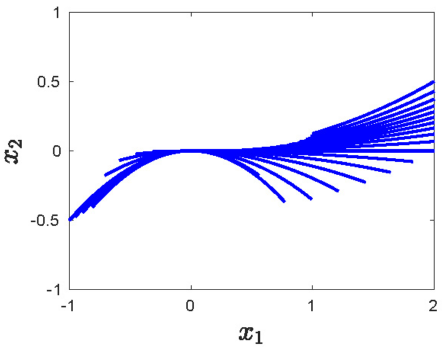
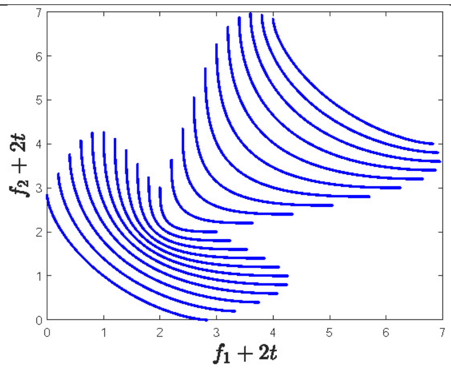
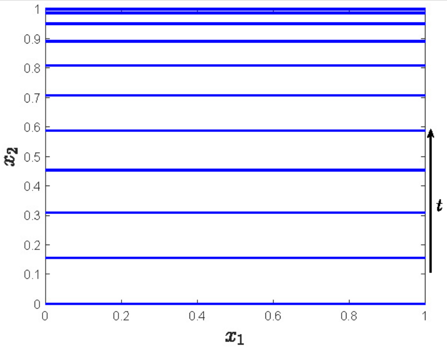
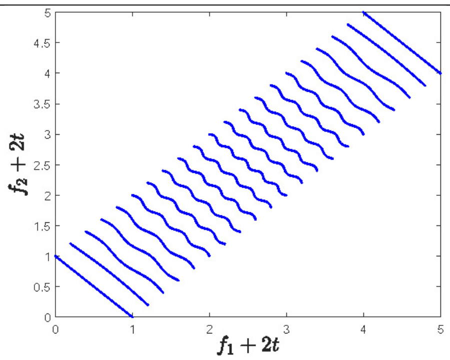
Problem	PS Illustration	PF Illustration	#objectives
DF1			2
DF2			2
DF3			2
DF4			2
DF5			2

TABLE II
ILLUSTRATIONS OF PS AND PF OF DF6-DF10 IN THE CEC 2018 BENCHMARK SUITE [9]

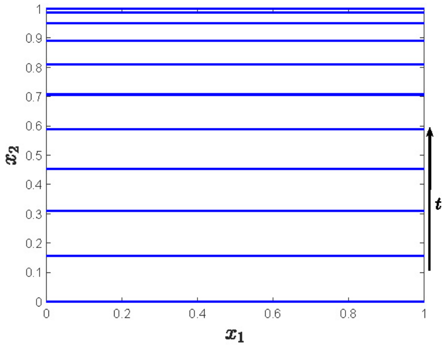
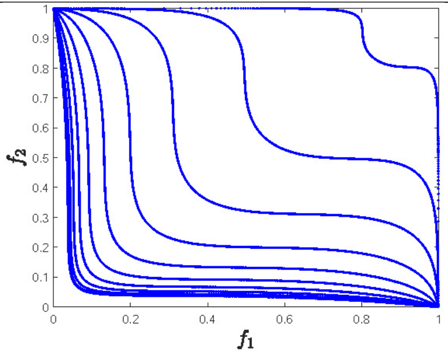
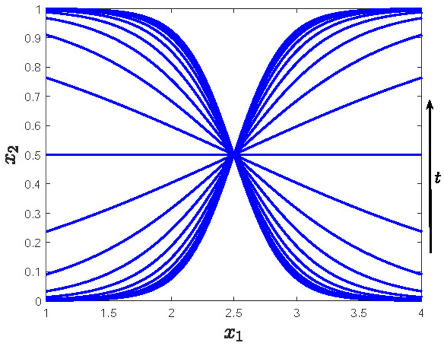
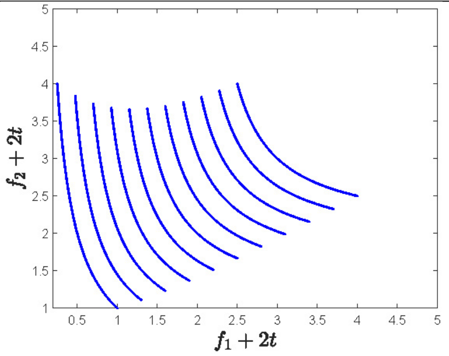
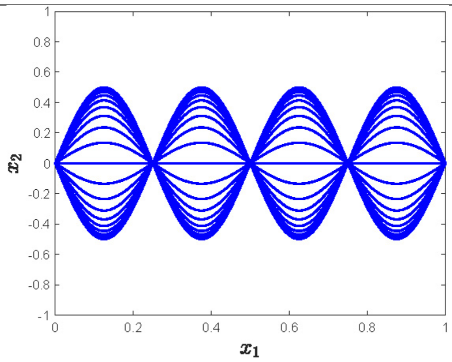
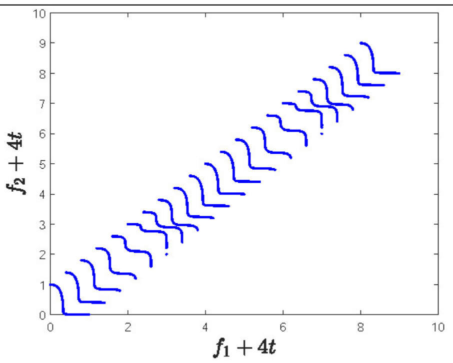
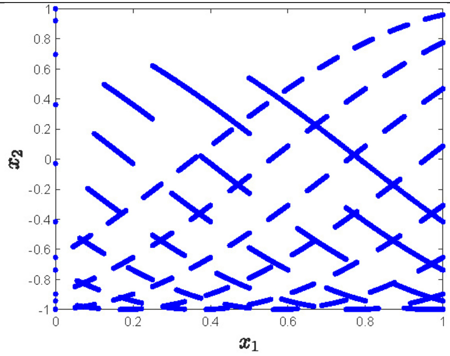
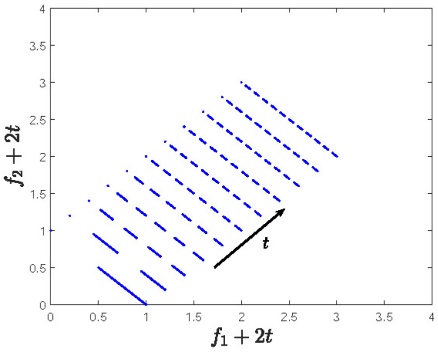
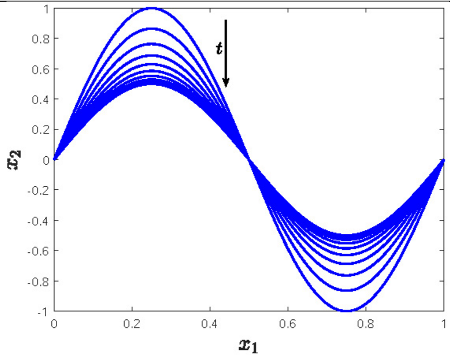
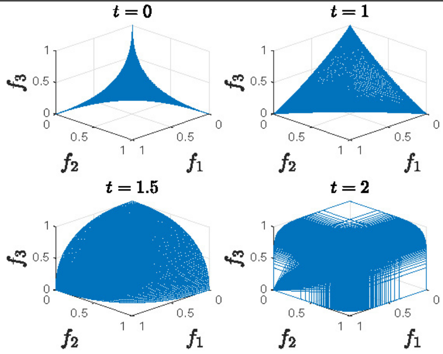
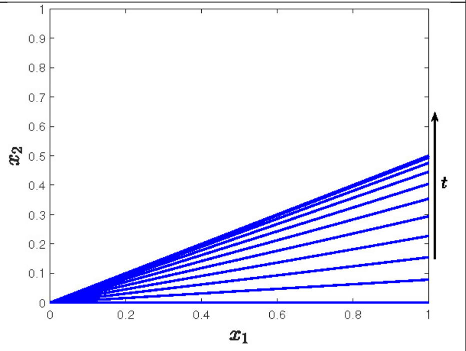
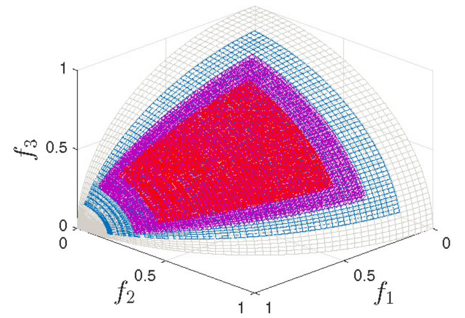
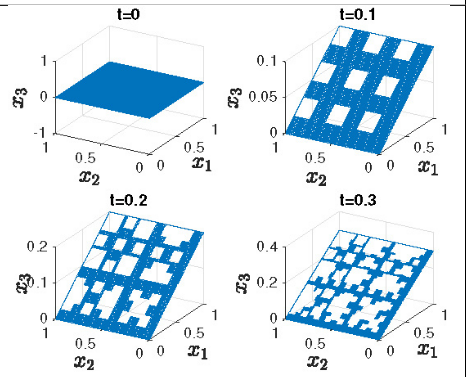
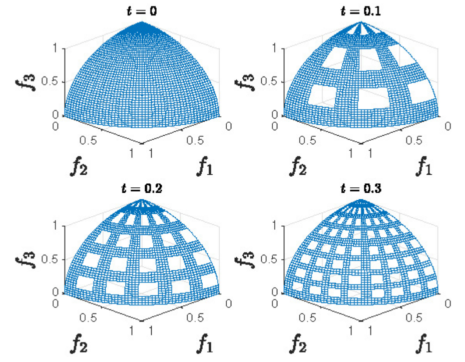
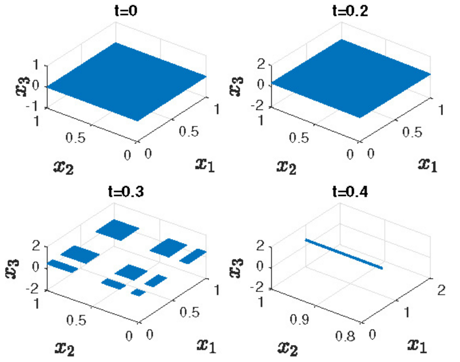
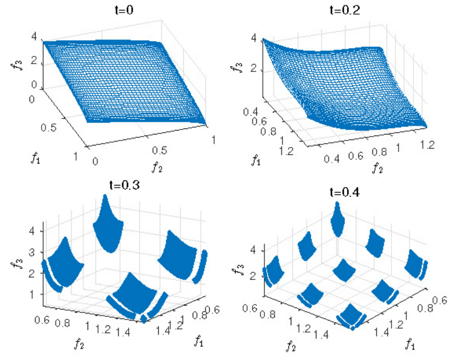
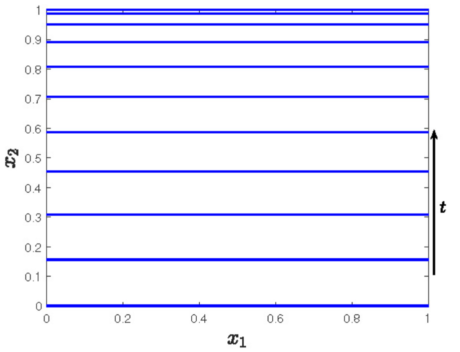
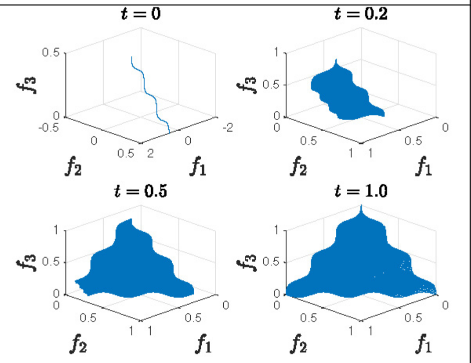
Problem	PS Illustration	PF Illustration	#objectives
DF6			2
DF7			2
DF8			2
DF9			2
DF10			3

TABLE III
ILLUSTRATIONS OF PS AND PF OF DF11-DF14 IN THE CEC 2018 BENCHMARK SUITE [9]

Problem	PS Illustration	PF Illustration	#objectives
DF11			3
DF12			3
DF13			3
DF14			3

the reference point and the solved non-dominated solution with objective function values of f. The Hypervolume Difference (HVD) quantifies the difference in hypervolume between the theoretical PF and the obtained PF. It is computed using Eq. (16), where $HV(PF)$ represents the hypervolume of the theoretical PF, and $HV(PF^{act})$ is the hypervolume of the solved PF.

$$HV(PF(t)) = \mathcal{L}(\cup_{f \in PF(t)} V_f) \qquad (15)$$

$$HVD(PF^{act}(t), PF(t)) = HV(PF^{act}(t)) - HV(PF(t)) \qquad (16)$$

In real-world scenarios, the theoretical PF is often unknown, whereas the lower bounds of each objective function in a DMOP can typically be determined with relative ease. Let $F^*(t) = (f_1^*(t), ..., f_{n(t)}^*(t))$ represent the lower bounds for each optimization objective in the DMOP. The Hypervolume Difference (HVD) formula is then adapted as shown in Eq. (17).

TABLE IV
PARAMETER SETTINGS IN EXPERIMENTS

Type	Parameter	Value
CEC 2018 Settings	change frequency (τ_t)	30
	change severity (n_t)	10, 30
	change number (T)	100
General Settings	number of key points	12
	number of independent experimental runs	20
	population size	100
NSGA-II Settings	mutation possibility	0.1
	crossover possibility	0.9

$$HVD^{act}(PF^{act}(t), PF(t)) = HV(PF^{act}(t)) - HV(F^{*}(t)) \tag{17}$$

MHVD is an enhancement of HVD that addresses its neglect of temporal or environmental dynamics. It is computed by averaging the HVD values over stable time intervals for each problem, as defined by Eq. (18):

$$MHVD = \frac{1}{T} \sum_{\tau \in [t_0, t_T]} HVD^{act}(PF^{act}(\tau), PS(\tau)) \tag{18}$$

To ensure statistical robustness, the performance evaluation experiments for each algorithm are repeated 20 independent times, and the mean values of each metric are compared in the following presented results to validate the proposed EnsembleFL introduced earlier. Furthermore, to assess the overall relative performance of the algorithms, we design a new metric based on all measured values as follows. First, given the experiment results, we can easily achieve the rank for each DMOEA in every metric and every DMOP. Then, for each metric and every DMOEA, we use the mean of ranks ($avgRank$) of all DMOPs to represent the overall performance in this metric, and average $avgRank$ ranks across all metrics to achieve the overall metric value. The calculations of $avgRank$ and the designed new metric are shown in Eq. (19) and (20). $\mathbb{A}$, $\mathbb{P}$ and $\mathbb{M}$ are the sets of DMOEAs (FLV, FLA, LR, SVR, XGB and EnsembleFL in this paper), DMOPs (DF1-DF14 in this paper) and metrics (MIGD and MHVD in this paper) used for the performance evaluation experiments, respectively. $m_{a,p}$ is the measured value of DMOEA a in the DMOP p and the metric m, which is the average of measured values by repeat experiments. $rank_{\mathbb{A}}(m_{a,p})$ represents the rank of the metric value achieved by a in all DMOEAs of $\mathbb{A}$, in the comparative experiments with p and m. $avgRank^{\mathbb{A}}_{a,m}$ is the mean of $rank_{\mathbb{A}}(m_{a,p})$ for all DMOPs, which indicates the overall performance of a in m. Then, by aggregating measured values in all performance metrics, $R^{\mathbb{A}}_{a}$ averages all ranks of $avgRank^{\mathbb{A}}_{a,m}$ with $m \in \mathbb{M}$ for a, which provides the relative overall performance of a in $\mathbb{A}$.

$$avgRank^{\mathbb{A}}_{a,m} = \frac{\sum_{p \in \mathbb{P}} rank_{\mathbb{A}}(m_{a,p})}{|\mathbb{P}|} \tag{19}$$

$$R^{\mathbb{A}}_{a} = \frac{\sum_{m \in \mathbb{M}} rank_{\mathbb{A}}(avgRank_{a,m})}{\mathbb{M}} \tag{20}$$

Tables II and III present the mean MIGD and MHVD values achieved by each algorithm on the CEC 2018 test problems, while Table IV summarizes the overall relative performance metrics of the algorithms. From Table II, it is evident that under severe environmental changes ($n_t = 10$), EnsembleFL achieves the best MIGD values on 10 DMOPs and the best MHVD values on 13 DMOPs. Table III demonstrates that under mild environmental changes ($n_t = 30$), EnsembleFL attains the best MIGD values on 5 DMOPs and the best MHVD values on 12 DMOPs. These results confirm that EnsembleFL exhibits superior adaptability to abrupt environmental shifts while maintaining high accuracy and diversity in its solution sets. Furthermore, Table IV reveals that EnsembleFL consistently outperforms all baseline algorithms across all metrics, achieving the highest overall performance ranking.

Tables II and III display the MIGD and MHVD values attained by each algorithm on the CEC 2018 test problems. Meanwhile, Table IV summarizes the overall relative performance metrics of the algorithms. As shown in Table II, it is clear that under severe environmental changes, EnsembleFL achieves the best MIGD values for 10 DMOPs and the best MHVD values for 13 DMOPs. Table III indicates that under mild environmental changes, EnsembleFL attains the best MIGD values for 5 DMOPs and the best MHVD values for 12 DMOPs. These results confirm that EnsembleFL demonstrates superior adaptability to abrupt environmental shifts while maintaining high accuracy and diversity in its solution sets. Furthermore, Table IV reveals that EnsembleFL consistently outperforms all baseline algorithms across all metrics, securing the highest overall performance ranking.

The exceptional performance of EnsembleFL stems from its integration of the FLA mechanism (designed to capture nonlinear environmental dynamics) and the ensemble framework. The efficacy of FLA is validated through direct comparisons with FLV. Tables II and III show that FLA outperforms FLV in approximately half of the experimental scenarios, while FLV prevails in the remaining cases. Table IV further highlights that under severe environmental changes, FLV and FLA exhibit comparable overall performance; however, under mild changes, FLA achieves better mean MIGD values, whereas FLV excels in mean MHVD values. This underscores the complementary strengths of different prediction mechanisms under varying environmental dynamics, justifying the necessity of the ensemble approach in EnsembleFL.

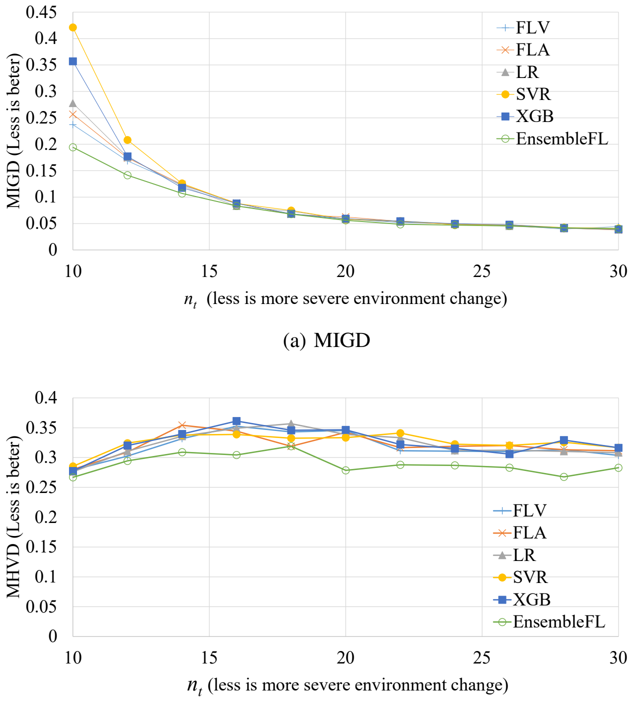
The outstanding performance of EnsembleFL originates from its integration of the FLA mechanism, which is designed to capture nonlinear environmental dynamics, and the ensemble framework. The effectiveness of the FLA mechanism is validated through direct comparisons with FLV. As illustrated in Tables II and III, FLA outperforms FLV in approximately half of the experimental scenarios, while FLV demonstrates superior performance in the remaining cases. Table IV further emphasizes that under severe environmental changes, FLV and FLA exhibit comparable overall performance. However, under mild changes, FLA achieves better MIGD values, whereas FLV excels in MHVD values. This highlights the complementary strengths of different prediction mechanisms under varying environmental dynamics, thereby justifying the necessity of the ensemble approach employed in EnsembleFL.

By integrating multiple prediction mechanisms, EnsembleFL leverages diverse environmental change patterns, achieving superior performance compared to individual predictors. As shown in Tables II and III, EnsembleFL surpasses both FLA and FLV in most scenarios, validating the effectiveness and efficiency of its ensemble strategy. In a few cases, EnsembleFL yields slightly worse MIGD values than FLV, primarily because linear environmental changes dominate these scenarios, where the integration of nonlinear predictions (from FLA) may interfere with FLV's linear extrapolation accuracy. In practice, if the environmental change pattern is known a priori, a specialized predictor (e.g., FLV for linear dynamics) would be ideal. However, real-world environments are characterized by heterogeneous and unpredictable changes, making general-purpose algorithms like EnsembleFL—capable of robust performance across diverse scenarios—far more practical and widely applicable.

By integrating multiple prediction mechanisms, EnsembleFL harnesses diverse environmental change patterns, thereby achieving superior performance compared to individual predictors. As demonstrated in Tables II and III, EnsembleFL outperforms both FLA and FLV in most scenarios, validating the effectiveness and efficiency of its ensemble strategy. In a limited number of cases, EnsembleFL yields slightly inferior MIGD values compared to FLV. This is primarily because linear environmental changes are predominant in these scenarios. In such cases, the integration of nonlinear predictions (from FLA) may potentially interfere with the linear extrapolation accuracy of FLV.

In practice, if the environmental change pattern is known in advance, a specialized predictor (e.g., FLV for linear dynamics) would be the optimal choice. However, real world environments are marked by heterogeneous and unpredictable changes. As a result, general-purpose algorithms like EnsembleFL, which are capable of delivering robust performance across a wide range of scenarios, are far more practical and widely applicable.

Next, we assess the performance stability of these DMOEAs by evaluating the performance trend with respect to the severity of environmental changes. In Fig. 2, we present the results for solving the DF5 problem. Similar performance change trends are observed when solving other DMOPs. From Fig. 2a, it can be observed that the accuracy of each DMOEA increases as the severity of environmental changes decreases. This is primarily because a DMOEA can perform more population evolutions in each stable environment of DMOPs with reduced severity, thereby increasing the likelihood of finding more and better solutions. In Fig. 2a, we can also observe the phenomenon where EnsembleFL achieves better MIGD values than other algorithms when the environmental changes are severe. This is because EnsembleFL makes more accurate predictions compared to others, largely due to the ensemble of multiple prediction methods that mine different PF moving patterns. However, the performance gaps between EnsembleFL and the other algorithms narrow when the environmental changes are mild (see the region where $n_t > 16$ in Fig. 2a). The main reason is that the impact of the evolutionary strategy on solving DMOPs becomes more significant as the severity of environmental changes decreases, since more evolution generations allow DMOEAs to explore a larger solution space. Therefore, in the future, we will focus on designing evolutionary strategies for DMOEAs to better adapt to solving DMOPs.



(a) MIGD

(b) MHVD

Fig. 2. The performance varied with the severity of environment changes in solving DF5.

As shown in Fig. 2b, every DMOEA has a small fluctuation in MHVD as the environment change severity varied, which represents that the solution diversity of searched PS is relatively stable. This may be because the solution diversity is largely decided by the population initialized for each stable environment, and thus depends on the prediction methods. By combining both results illustrated in Fig. 2, we can make a conclusion that the quality (accuracy and diversity) of solutions achieved by a DMOEA is determined by both the prediction method adapt to the environment changes and the evolutionary strategy that provides efficient and effective search directions. EnsembleFL achieves better MHVD than other algorithms in all cases, shown in Fig. 2b, which further verifies its superior performance.

Now, we conduct experiments to evaluate the performance changes of DMOEAs when the problem scale of the solved DMOP is varied. We also take DF5 as a case, and illustrate the results in Fig. 3. From Fig. 3a, we can see that as the problem scale (the variable number) is increased, the accuracy of solutions is decreased for every DMOEA. This is mainly because the solution space is expanded greatly as the problem scale is increased, which would make the solution search much more difficult. As shown in Fig. 3a, EnsembleFL can always achieve the best MIGD in all different problem scale cases, and the

TABLE V
PERFORMANCE METRIC VALUES OF ALGORITHMS UNDER SEVERE ENVIRONMENTAL CHANGES ($n_t = 10$) ON THE CEC 2018 BENCHMARK SUITE.

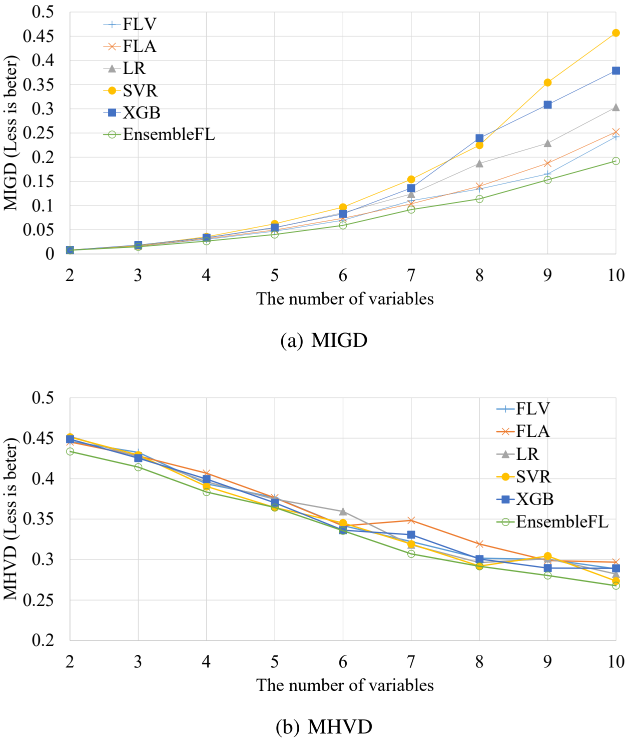
DMOP	Metric		FLV	FLA	LR	SVR	XGB	EnsembleFL
DF1	**MIGD**	**average**	0.0033583	0.0034201	0.0036141	0.0036728	0.0033457	**0.003286**
		rank	3	4	5	6	2	**1**
	MHVD	**average**	0.5721015	0.5722917	0.5723873	0.5727552	0.5719026	**0.571343**
		rank	3	4	5	6	2	**1**
DF2	**MIGD**	**average**	0.2756588	0.2796511	0.2670932	0.2620286	0.2734191	**0.220203**
		rank	5	6	3	2	4	**1**
	MHVD	**average**	0.500763	0.4999689	0.4868519	0.4987084	0.4948678	**0.469391**
		rank	6	5	2	4	3	**1**
DF3	**MIGD**	**average**	0.6879658	0.6834649	0.6720004	0.6450547	0.6930142	**0.616694**
		rank	5	4	3	2	6	**1**
	MHVD	**average**	0.9331428	0.940332	0.8956523	0.8416667	0.9205133	**0.826622**
		rank	5	6	3	2	4	**1**
DF4	**MIGD**	**average**	0.4063925	0.4378698	0.4303647	0.3755076	0.3801695	0.3903407
		rank	4	6	5	1	2	3
	MHVD	**average**	0.2370486	0.2250237	0.2234691	**0.212787**	0.2236281	0.2368965
		rank	6	4	2	1	3	5
DF5	**MIGD**	**average**	0.2300044	0.258425	0.2879263	0.4319601	0.3671064	**0.201374**
		rank	2	3	4	6	5	**1**
	MHVD	**average**	0.2836831	0.2949004	0.2741977	0.2931181	0.2889677	**0.267597**
		rank	3	6	2	5	4	**1**
DF6	**MIGD**	**average**	9.5588982	9.0174784	**9.003745**	10.218425	10.380135	9.7105436
		rank	3	2	1	5	6	4
	MHVD	**average**	0.1686466	0.171978	0.2274755	0.2169132	0.2448779	**0.146315**
		rank	2	3	5	4	6	**1**
DF7	**MIGD**	**average**	0.409614	0.4131103	0.3642087	0.4374911	0.40696	**0.343451**
		rank	4	5	2	6	3	**1**
	MHVD	**average**	0.8182154	0.8205052	0.7967074	0.8199414	0.8236905	**0.773039**
		rank	3	5	2	4	6	**1**
DF8	**MIGD**	**average**	0.2197897	0.1968333	0.2244304	0.2003511	0.1985181	**0.14855**
		rank	5	2	6	4	3	**1**
	MHVD	**average**	0.4778251	0.4365768	0.4820203	0.4785657	0.4742142	**0.327993**
		rank	4	2	6	5	3	**1**
DF9	**MIGD**	**average**	1.204431	1.1780366	1.1953921	1.2896327	1.3823071	**0.845348**
		rank	4	2	3	5	6	**1**
	MHVD	**average**	0.3083181	0.3122055	0.3127468	0.3246054	0.3302476	**0.26604**
		rank	2	3	4	5	6	**1**
DF10	**MIGD**	**average**	**0.35589**	0.388526	0.4130915	0.3948649	0.3874044	0.3836652
		rank	1	4	6	5	3	2
	MHVD	**average**	0.2399545	0.235306	0.2130403	0.2496932	0.2284013	**0.172932**
		rank	5	4	2	6	3	**1**
DF11	**MIGD**	**average**	**11.79304**	11.822246	11.825885	11.83362	11.809165	11.814771
		rank	1	4	5	6	2	3
	MHVD	**average**	0.7217686	0.7162597	0.7227602	0.7211256	0.720429	**0.712115**
		rank	5	2	6	4	3	**1**
DF12	**MIGD**	**average**	0.5292982	0.5199371	0.5076853	0.5244829	0.5105333	**0.478483**
		rank	6	4	2	5	3	**1**
	MHVD	**average**	0.204682	0.1958073	0.2083629	0.2078805	0.1910199	**0.140718**
		rank	4	3	6	5	2	**1**
DF13	**MIGD**	**average**	1.1431864	1.0961584	1.1435138	1.2622611	1.2865144	**1.073832**
		rank	3	2	4	5	6	**1**
	MHVD	**average**	0.1624251	0.1462736	0.1560467	0.1722161	0.1626285	**0.126949**
		rank	4	2	3	6	5	**1**
DF14	**MIGD**	**average**	0.3247037	0.3684833	0.3961349	0.5067011	0.4323082	**0.296605**
		rank	2	3	4	6	5	**1**
	MHVD	**average**	0.2768491	0.2947282	0.2771067	0.3091701	0.3101942	**0.267537**
		rank	2	4	3	5	6	**1**

TABLE VI
PERFORMANCE METRIC VALUES OF ALGORITHMS UNDER MILD ENVIRONMENTAL CHANGES ($n_t = 30$) ON THE CEC 2018 BENCHMARK SUITE.

DMOP	Metric		FLV	FLA	LR	SVR	XGB	EnsembleFL
DF1	**MIGD**	**average**	**0.003189**	0.0032049	0.003193	0.003233	0.0032442	0.0032387
		rank	**1**	3	2	4	6	5
	MHVD	**average**	0.6337584	0.6335523	0.6337234	0.6336245	0.6337459	**0.633462**
		rank	6	2	4	3	5	**1**
DF2	**MIGD**	**average**	0.0322955	0.0329503	0.0336411	**0.029833**	0.0309913	0.030763
		rank	4	5	6	**1**	3	2
	MHVD	**average**	0.3393667	0.3458686	0.3355523	0.3394374	0.3414094	**0.331918**
		rank	3	6	2	4	5	**1**
DF3	**MIGD**	**average**	0.304369	0.3138129	0.5708436	0.5103331	0.5865344	**0.248663**
		rank	2	3	5	4	6	**1**
	MHVD	**average**	0.682444	0.6936069	0.927243	0.8975001	0.9710108	**0.564021**
		rank	2	3	5	4	6	**1**
DF4	**MIGD**	**average**	0.1546621	0.1645479	0.1657822	0.1828026	0.1746745	**0.150169**
		rank	2	3	4	6	5	**1**
	MHVD	**average**	0.1894664	0.1892574	**0.18718**	0.1874358	0.1904365	0.1894794
		rank	4	3	**1**	2	6	5
DF5	**MIGD**	**average**	0.0408911	0.0380041	0.0421934	0.039994	0.0425632	**0.037595**
		rank	4	2	5	3	6	**1**
	MHVD	**average**	0.3093846	0.3177793	0.3265994	0.2984409	0.3082299	**0.278688**
		rank	4	5	6	2	3	**1**
DF6	**MIGD**	**average**	**2.805715**	3.1042128	3.8518684	3.4743493	2.9011315	2.8421219
		rank	**1**	4	6	5	3	2
	MHVD	**average**	0.3594155	0.3520524	0.3877135	0.3915496	0.3422279	**0.258814**
		rank	4	3	5	6	2	**1**
DF7	**MIGD**	**average**	0.2957322	0.272228	0.2889933	**0.266037**	0.2690157	0.2983929
		rank	5	3	4	**1**	2	6
	MHVD	**average**	0.7347463	0.724187	0.7302869	**0.716875**	0.723876	0.7372737
		rank	5	3	4	**1**	2	6
DF8	**MIGD**	**average**	0.1551293	0.1718291	0.1840825	0.1780454	0.1759159	**0.146287**
		rank	2	3	6	5	4	**1**
	MHVD	**average**	0.3863491	0.4344777	0.4450676	0.4491391	0.4482893	**0.32062**
		rank	2	3	4	6	5	**1**
DF9	**MIGD**	**average**	0.1910382	0.1821445	0.2029375	0.188856	0.2176944	**0.154081**
		rank	4	2	5	3	6	**1**
	MHVD	**average**	0.362736	0.3520702	0.3587474	0.3698268	0.3636772	**0.273715**
		rank	4	2	3	6	5	**1**
DF10	**MIGD**	**average**	0.1362368	0.1367703	0.1290924	0.1352061	**0.127096**	0.1521059
		rank	4	5	2	3	**1**	6
	MHVD	**average**	0.0051094	0.0048734	0.0041344	0.0046356	0.0040479	**0.003771**
		rank	6	5	3	4	2	**1**
DF11	**MIGD**	**average**	11.924175	**11.90482**	11.91003	11.919173	11.92976	11.905475
		rank	5	**1**	3	4	6	2
	MHVD	**average**	0.7435284	0.7432473	0.7392181	0.7422711	0.7472291	**0.728913**
		rank	5	4	2	3	6	**1**
DF12	**MIGD**	**average**	**0.189942**	0.2070001	0.1938938	0.2076459	0.2058826	0.1903504
		rank	**1**	5	3	6	4	2
	MHVD	**average**	0.122213	0.115762	0.1274722	0.126609	0.1189187	**0.089405**
		rank	4	2	6	5	3	**1**
DF13	**MIGD**	**average**	0.2838913	**0.271139**	0.285401	0.2836665	0.2867742	0.2859133
		rank	3	**1**	4	2	6	5
	MHVD	**average**	0.1272296	0.1364537	0.1230229	0.1264431	0.1237361	**0.106456**
		rank	5	6	2	4	3	**1**
DF14	**MIGD**	**average**	0.0871841	0.0869186	**0.086192**	0.0866514	0.0869227	0.0862841
		rank	6	4	**1**	3	5	2
	MHVD	**average**	0.3266595	0.3216569	0.3292095	0.322871	0.3177873	**0.307644**
		rank	5	3	6	4	2	**1**

TABLE VII
THE OVERALL PERFORMANCE COMPARISON OF ALGORITHMS.

Environment	Metric		FLV	FLA	LR	SVR	XGB	EnsembleFL
Severe change ($n_t = 10$)	**MIGD**	$avgRank$	3.43	3.64	3.79	4.57	4	**1.57**
		rank	2	3	4	6	5	**1**
	MHVD	$avgRank$	3.86	3.79	3.64	4.43	4	**1.29**
		rank	4	3	2	6	5	**1**
	Overall	R	3	3	3	6	5	**1**
		rank	2	2	2	6	5	**1**
Mild change ($n_t = 30$)	**MIGD**	$avgRank$	3.14	3.14	4	3.57	4.5	**2.64**
		rank	2	2	5	4	6	**1**
	MHVD	$avgRank$	4.21	3.57	3.79	3.86	3.93	**1.64**
		rank	6	2	3	4	5	**1**
	Overall	R	4	2	4	4	5.5	**1**
		rank	3	2	3	3	6	**1**



(a) MIGD

(b) MHVD

Fig. 3. The performance varied with the problem scale (the variable number) in solving DF5.

performance differences of EnsembleFL to others are becoming larger when the problem scale is increased. This illustrates that EnsembleFL perform better on solving DMOPs than other algorithms. In Fig. 3b, we can see that the performance value is getting better in MHVD as the problem scale increasing for each algorithm. This is mainly because the dimension of the solution space is increased with the problem scale, which can result in an decreasing HVD between actual PS and theoretical PS as their difference is generally less than 1 unit in each dimension. EnsembleFL achieves the best MIGD and MHVD in all of DMOEAs and all problem scales, as illustrated in Fig. 3. These results confirm the superior performance of our proposed algorithm in solving DMOPs again.

V. RELATED WORK

The inherent uncertainty of environmental changes poses significant challenges in solving DMOPs. To address these challenges, many studies focus on reinitializing populations in DMOEAs during environmental changes, aiming to rapidly adapt to changes and accelerate the discovery of the optimal solutions. The simplest approach involves reinitializing part or all of the population to enhance diversity and avoid local optima [8]. While computationally efficient, this method treats each DMOP phase as an independent Multi-Objective Problem (MOP), ignoring historical change patterns and resulting in poor search efficiency.

To improve this, memory-based environmental response mechanisms have been proposed. For example, some works retrieve solutions from a memory bank when encountering similar historical environments and combine them with randomly initialized individuals [16]. However, in highly dynamic DMOPs, environmental similarity is often low, and robust similarity detection remains challenging, limiting the utility of such memory-based strategies. Therefore, a multi-scenario modeling approach [17] was introduced to categorize historical environments into clusters, using uniformly sampled reference solutions within each cluster. When a new environment arises, the closest cluster is identified, and its reference solutions are reused. If no similar cluster exists, a new one is created and solved via conventional optimization. Similarity is measured using fuzzy proximity metrics based on decision-space sampling.

For high-dimensional DMOPs, memory-based methods often underperform due to low inter-environment similarity. Consequently, prediction-based strategies have emerged, leveraging historical PS evolution patterns to forecast PS trajectories and generate solutions that accelerate convergence. For instance, reference [10] generates sub-populations via three strategies: (1) linear extrapolation of PS centroids, (2) selection of elite individuals, and (3) random initialization. During evolution, the first strategy generates offspring to enhance convergence. Reference [18] adaptively updates decision variables based on their classification as “convergence-sensitive” or “diversity-sensitive”, applying controlled mutations or region-based adjustments. Reference [19] uses autoregression to predict centroid movements and Gaussian perturbations to diversify populations.

Most methods assume linear PS changes, leading to suboptimal performance in nonlinear scenarios. To address this, Reference [20] classifies decision variables as linear/nonlinear via correlation coefficients, applying Lagrange extrapolation or Fourier-based autoregression. Reference [21] targets rotational DMOPs by incorporating rotation angles into gradient predictions. Reference [22] combines linear (centroid-based) and nonlinear (angle-based) models for dynamic Regions of Interest (ROIs). Reference [23] exploits Gaussian process regression, linear moving characteristic of knee point-based partitions and the historical Pareto-similarity to predict new positions of PS.

Recent advances integrate machine learning (ML) for enhanced prediction. Reference [11] trains SVM classifiers on candidate sets (current population, offspring generated by differential evolution, and pivot points) to select initial populations, supplemented by MOEA/D evolution. Reference [13] employs XGBoost to predict individual movements based on historical trajectories. Reference [24], [25] uses RNNs/LSTMs to forecast PS dynamics, with adaptive population replacement to mitigate early-stage prediction errors. Despite their potential, ML methods face challenges such as high computational overhead and sensitivity to training data quality.

Most approaches rely on single prediction mechanisms, excelling only in specific change modes (e.g., linear, rotational) while lacking universality. To overcome this, our work introduces an ensemble learning framework that integrates multiple predictors (e.g., FLV, FLA) to handle diverse environmental dynamics. By synergizing linear and nonlinear prediction models, our method enhances prediction accuracy across varied scenarios, improving DMOEA convergence and diversity in dynamic environments.

VI. Conclusion

This study explores evolutionary algorithms for solving DMOPs. To accommodate the diversity of environmental change patterns in DMOPs, we propose a novel ensemble framework inspired by the Bagging ensemble learning approach. This framework integrates two feed-forward prediction mechanisms, namely FLV and FLA, to enhance prediction accuracy across diverse environmental scenarios, thereby improving the exploration-exploitation balance of DMOEAs. The proposed method is extensively evaluated on 14 DMOPs from the CEC 2018 benchmark suite. The results demonstrate that the algorithm attains superior accuracy and generates well-distributed solution sets under dynamic conditions.

This work confirms the effectiveness of the ensemble prediction strategy. Future research will focus on two key directions:

1) Expanding Prediction Mechanisms: exploring the integration of additional advanced prediction techniques (e.g., hybrid models or deep learning-based predictors) to further refine solution set accuracy.
2) Enhancing Evolutionary Strategies: Designing more efficient evolutionary operators and adaptation mechanisms to strengthen the global search capability of DMOEAs, improving the accuracy, diversity, and convergence rate in searching PF.

References

[1] L. Barruffo, B. Caiazzo, A. Petrillo, and S. Santini, “A goa4 control architecture for the autonomous driving of high-speed trains over etcs: Design and experimental validation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 5096–5111, 2024.

[2] L. Xin, W. Zegui, Z. Ruqing, and L. Fusheng, “Dynamic multi-agent deep deterministic policy gradient for autonomous navigation of reconfigurable unmanned aerial vehicle,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 10 574–10 580.

[3] L. Zhao, Y. Bai, and J. K. Paik, “Achieving optimal-dynamic path planning for unmanned surface vehicles: A rational multi-objective approach and a sensory-vector re-planner,” *Ocean Engineering*, vol. 286, p. 115433, 2023.

[4] H.-B. Zhang and B. Huang, “Research on mobile robot path planning based on multi-strategy improved ant colony optimization algorithm,” *Engineering Letters*, vol. 33, no. 3, pp. 688–703, 2025.

[5] Y.-B. Wang, J.-S. Wang, and X.-F. Sui, “Improved particle swarm optimization algorithm with logistic function and trigonometric function for three-dimensional path planning problems,” *Engineering Letters*, vol. 33, no. 2, pp. 442–459, 2025.

[6] R. Wu, C. Dai, S. Tian, X. Wang, and T. Wang, “A bi-objective optimization model for urban rail transit train timetable under uncertain passenger demand,” *IAENG International Journal of Applied Mathematics*, vol. 55, no. 3, pp. 500–513, 2025.

[7] G. Yuan, X. Liu, C. Zhu, C. Wang, M. Zhu, and Y. Sun, “Multi-objective coupling optimization of electrical cable intelligent production line driven by digital twin,” *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102682, 2024.

[8] K. Deb, U. B. R. N., and S. Karthik, “Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling,” in *Proceedings of the 4th international conference on Evolutionary multi-criterion optimization*, Matsushima, Japan, 2007, pp. 803–817.

[9] S. Jiang, S. Yang, X. Yao, K. Tan, M. Kaiser, and N. Krasnogor, “Benchmark problems for cec’2018 competition on dynamic multiobjective optimisation,” pp. 1–18, December 2017. [Online]. Available: http://homepages.cs.ncl.ac.uk/shouyong.jiang/cec2018/CEC2018_Tech_Rep_DMOP.pdf

[10] Q. Li, X. Liu, F. Wang, S. Wang, P. Zhang, and X. Wu, “A framework based on generational and environmental response strategies for dynamic multi-objective optimization,” *Applied Soft Computing*, vol. 152, p. 111114, 2024.

[11] Z. Fang, H. Li, L. Hu, and N. Zeng, “A learnable population filter for dynamic multi-objective optimization,” *Neurocomputing*, vol. 574, p. 127241, 2024.

[12] H. Sun, X. Ma, Z. Hu, J. Yang, and H. Cui, “A two stages prediction strategy for evolutionary dynamic multi-objective optimization,” *Applied Intelligence*, vol. 53, no. 1, pp. 1115–1131, 2023.

[13] K. Gao and L. Xu, “Novel strategies based on a gradient boosting regression tree predictor for dynamic multi-objective optimization,” *Expert Systems with Applications*, vol. 237, p. 121532, 2024.

[14] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, “Modified distance calculation in generational distance and inverted generational distance,” in *8th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2015)*, Guimarães, Portugal, 2015, pp. 110–125.

[15] S. Jiang, M. Kaiser, S. Yang, S. Kollias, and N. Krasnogor, “A scalable test suite for continuous dynamic multiobjective optimization,” *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2814–2826, 2020.

[16] X. Wang and sanyi Li, “Dynamic multi-objective optimization algorithm based on density and memory,” *Computer Simulation*, vol. 39, no. 9, pp. 49–54, 2022.

[17] B. Xu, X. Lv, W. Li, Z. Fan, D. Gong, and J. He, “Dynamic robust multi-objective evolutionary optimization algorithm based on multi-scenario modeling,” *Control and Decision*, vol. 39, no. 12, pp. 3997–4006, 2024.

[18] Z.-P. Liang, H.-C. Li, Z.-Q. Wang, K.-F. Hu, and Z.-X. Zhu, “Dynamic multi-objective evolutionary algorithm with adaptive change response,” *Acta Automatica Sinica*, vol. 49, no. 8, pp. 1688–1706, 2023.

[19] A. Zhou, Y. Jin, and Q. Zhang, “A population prediction strategy for evolutionary dynamic multiobjective optimization,” *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014.

[20] F. Min, W. Dong, and W. Ding, "Dynamic multi-objective evolutionary algorithm based on classification of decision variable temporal change characteristics," *Acta Automatica Sinica*, vol. 50, no. 11, pp. 2154–2176, 2024.

[21] E. Li and C. Liu, "Classification multi-strategy predictive dynamic multi-objective optimization with pareto set rotation," *Computer Engineering and Applications*, vol. 60, no. 22, pp. 87–104, 2024.

[22] X. Wang, J. Zheng, Z. Hou, Y. Liu, J. Zou, Y. Xia, and S. Yang, "A novel preference-driven evolutionary algorithm for dynamic multi-objective problems," *Swarm and Evolutionary Computation*, vol. 89, p. 101638, 2024.

[23] Y. Hu, Y. Li, J. Ou, J. Peng, J. Li, and J. Zheng, "Dynamic multi-objective evolutionary algorithm based on dual-layer collaborative prediction under multiple perspective," *Swarm and Evolutionary Computation*, vol. 94, p. 101876, 2025.

[24] Y. Hu, J. Ou, P. N. Suganthan, W. Pedrycz, R. Wang, J. Zheng, J. Zou, and Y. Song, "Dynamic multi-objective optimization algorithm guided by recurrent neural network," *IEEE Transactions on Evolutionary Computation (In Press)*, pp. 1–14, 2024.

[25] G. Chen and Y. Guo, "A lstm assisted prediction strategy for evolutionary dynamic multiobjective optimization," in *4th International Conference on Neural Computing for Advanced Applications (NCAA 2023)*, Hefei, China, 2023, pp. 376–389.