Evolutionary Dynamic Multi-Objective Optimization with Learning across Problems

Anonymous Authors

***Abstract*—Dynamic multi-objective optimization problems (DMOPs) are prevalent in many practical applications and have garnered significant attention from both industry and academia, leading to the proposal of numerous dynamic multi-objective optimization algorithms. Among these approaches, learning- and prediction-based evolutionary approaches have achieved remarkable success due to their fast learning and strong optimization capabilities when dynamic occurs. However, existing methods typically focus on learning and transferring knowledge within a single problem, often relying on historical knowledge to aid the optimization process. This could limit the data available for developing a more general prediction model for dynamic optimization. The potential for leveraging knowledge across different DMOPs to enhance problem-solving efficiency and effectiveness remains largely unexplored. Building on this insight, this paper explores the solution of DMOPs with learning not only within a single problem but also across different problems. In particular, by performing dynamic feature extraction and task-specific solution classification, we propose to construct a centralized learning model that captures the correlations across DMOPs and the corresponding optimized solutions. This approach allows optimization data from multiple problems to be effectively utilized, enabling the learning of dynamic knowledge to enhance evolutionary optimization when a change occurs. To assess the effectiveness of the proposed algorithm, extensive empirical studies have been conducted on the commonly used DMOP benchmarks with fixed and shifty dynamic settings. The numerical results demonstrate the effectiveness of the proposed algorithm in recognizing more general change patterns of DMOPs, showcasing its superiority compared to the existing state-of-the-art learning and prediction-based approaches.**

***Index Terms*—Dynamic multi-objective optimization, Evolutionary algorithms, Learning and prediction-based method, Centralized learning model.**

I. Introduction

DYNAMIC multi-objective optimization (DMO) is a crucial research area that has received significant attention due to its wide applicability in solving complex real-world dynamic multi-objective optimization problems (DMOPs), such as industrial scheduling [1], motion planning of robots [2], system control [3], [4] and optimization [5]. These problems, characterized by their evolving objectives and constraints over time, pose unique challenges that demand innovative solutions [6], [7]. The development and refinement of optimization algorithms capable of dynamically adapting to changes is of paramount importance in both industrial applications and academic research [8], [9].

Traditionally, the field of DMO has witnessed the formulation and application of numerous algorithms designed to tackle the dynamic nature of these problems. Among the myriad of approaches, evolutionary methods based on learning and prediction have emerged as particularly effective. These algorithms capitalize on the ability to learn from past data and predict future states, thereby facilitating rapid and robust evolutionary optimization adjustments in response to dynamic changes. For instance, Jiang et al. [10] utilized a two-stage search mechanism, employing a pre-search strategy to filter low-quality individuals and a Transfer Adaboost (TrAdaboost) classifier to forecast the optimal initial population. Afterward, Jiang et al. [11] devised a linear prediction model for knee point movements, combining a support vector machine with TrAdaboost to form an unbalanced classification framework. This model aimed to transfer minimal high-quality knowledge and accelerate the convergence of search. Furthermore, Liu et al. [12] proposed an interaction-based prediction method that models the correlation between variables and prediction targets. This method selects the correlation variables to construct the prediction model to forecast solutions for evolutionary DMO when changes occur. More recently, Guo et al. [13] proposed a transfer-knowledge-guided evolutionary DMO strategy. This method extracts and preserves knowledge as twin-tuples, removes redundancies, and evaluates stored knowledge in new environments using a knowledge-matching strategy.

However, a critical examination of existing learning and prediction-based approaches reveals a significant limitation: their focus is predominantly inward, concentrating on knowledge acquisition and transfer within the confines of a single problem [14], [15], [16], [17]. The inward focus leverages historical data from the problem itself, primarily before any dynamic changes, to inform and guide the optimization process post-change. Such a strategy, while useful, inherently restricts the scope of learning from the data of a singular problem context. The potential of a broader and more inclusive approach that extends learning and predictive capabilities across different DMOPs remains largely untapped. By adopting a more holistic view that encompasses data and insights from various problems, there is a compelling opportunity to enhance the effectiveness and efficiency of optimization algorithms in dynamic environments [18], [19], [20].

Keeping in mind the above observations, this paper embarks on a study to explore a centralized learning and prediction model to capture generalized knowledge in solving DMOPs. This model is designed to provide useful guidance to the evolutionary optimization process, even when the first dynamic occurs in a given dynamic problem. This is impractical for existing learning and prediction evolutionary approaches, as they rely on historical data from the same DMOP. To the best

of our knowledge, this work represents the first attempt in the literature to explore evolutionary DMO across different problems. Specifically, to leverage cross-domain knowledge and enable learning from solved problems to optimized solutions, we first construct a neural network-based shared extractor of the centralized learning model to extract and fuse dynamic features of solved problems. Subsequently, the archived optimal solutions with merged features are inputted to the task-specific classifiers to realize the learning of dynamic optimization patterns, and finally, the guiding solutions suggested by the learning model can be used to enhance the evolutionary optimization of the encountered problem after a dynamic occurs. To assess the performance of the proposed algorithm, comprehensive empirical studies are conducted on the widely adopted DMOP benchmark suite under both steady and fluctuating dynamic settings. The experimental results obtained from the comparison with state-of-the-art learning and prediction-based DMO algorithms demonstrate its superiority and significance in achieving more general and effective optimization performance via learning across problems. In summary, the threefold contributions of this paper are highlighted as follows:

- Propose to investigate and harness the potential of leveraging knowledge across dynamic multi-objective problems to enhance the efficiency and effectiveness of problem-solving.
- Develop an end-to-end centralized learning model to capture the relationships across different dynamic multi-objective optimization problems and their corresponding optimized solutions.
- Present comprehensive comparative results on the widely adopted DMOP test suite with fixed and shifty dynamic configurations, demonstrating that the proposed algorithm exhibits competitive superiority over existing state-of-the-art learning and prediction-based approaches.

The remainder of this research is structured as follows: Section II introduces the definition of the target DMOP and reviews existing learning and prediction-based algorithms for solving DMOPs. In Section III, the details of our proposed dynamic multi-objective optimization with learning across problems are elaborated. Section IV outlines the specific experimental designs, including original and modified benchmark settings, evaluation metrics, and the algorithms used for comparison. Following this, Section V presents comprehensive empirical studies on commonly adopted DMOP benchmarks, covering both fixed and shifting dynamic configurations. Finally, Section VI concludes the paper with key remarks and discusses potential directions for future research.

II. Preliminaries

This section starts with a preliminary introduction to the DMOP, followed by an overview of existing evolutionary algorithms to solve DMOPs.

A. Dynamic Multi-objective Optimization

By discretizing time, a DMOP can be viewed as a series of stationary multi-objective optimization problems. Taking a minimization problem as an instance, a typical DMOP is mathematically expressed as follows:

$$\begin{aligned} \min \quad & F(\mathrm{x},t) = [f_1(\mathrm{x},t), \cdots, f_m(\mathrm{x},t)]^T \\ s.t. \quad & \mathrm{x} \in \Omega = \prod_{i=1}^{n}[L_i, U_i], G(\mathrm{x},t) \leq 0, H(\mathrm{x},t) = 0 \end{aligned} \tag{1}$$

where $F(\mathrm{x},t)$ is the set of objectives in terms of discrete time or environment, which is indexed by t. The vector $\mathrm{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ comprises n decision variables within the decision space $\Omega \subset \mathbb{R}^n$. The given DMOP encompasses a multidimensional objective space $\mathbb{R}^m$ with m-number of objectives. Each objective function, denoted by f_i, is continuous with respect to x. The variable x_i is constrained within the range $[L_i, U_i]$. Additionally, G and H represent the inequality and equality constraints, respectively.

Moreover, the primary goal of a DMO algorithm is to promptly track the shifting Pareto-Optimal Set (POS) or Pareto-Optimal Front (POF) in response to environmental changes. At time or environment t, a decision vector x_1 is Pareto dominated by another vector x_2, denoted by $\mathrm{x}_1 \prec_t \mathrm{x}_2$, if and only if:

$$\begin{cases} \forall i \in \{1, \ldots, m\} & f_i(\mathrm{x}_1, t) \geq f_i(\mathrm{x}_2, t) \\ \exists i \in \{1, \ldots, m\} & f_i(\mathrm{x}_1, t) > f_i(\mathrm{x}_2, t) \end{cases} \tag{2}$$

According to the dynamic Pareto dominance concept, the dynamic POS, dynamic POF, as well the frequency and severity of dynamic are defined as follows [6], [7]:

***Definition* 1** (Dynamic Pareto Optimal Solutions)**.** *The POS at t in the decision space, denoted as $DPOS(t)^*$, is the set of current Pareto optimal solutions such that:*

$$DPOS(t)^* = \{\mathbf{x}_i^* | \nexists f(\mathbf{x}_j, t) \prec f(\mathbf{x}_i^*, t)^*, f(\mathbf{x}_j, t) \in F^m\} \tag{3}$$

***Definition* 2** (Dynamic Pareto Optimal Front)**.** *The POF at t, denoted as $DPOF(t)^*$, is the corresponding projection of the $DPOS(t)^*$ in the objective space such that:*

$$DPOF(t)^* = \{f(\mathbf{x}_i, t)^* | \nexists f(\mathbf{x}_j, t) \prec f(\mathbf{x}_i, t)^*, f(\mathbf{x}_j, t) \in F^m\} \tag{4}$$

***Definition* 3** (The Frequency of Dynamic)**.** *The frequency of dynamic is denoted as $\frac{1}{\tau_t}$, which measures how frequently a DMOP changes. In dynamic optimization, τ_t can be either specified as the counter of fitness evaluations or iterative generations.*

***Definition* 4** (The Severity of Dynamic)**.** *The severity of the dynamic is denoted as n_t, weighing the magnitude of a dynamic change or perturbation in a stationary environment. It can be mathematically defined as the amount of dynamic causing the shift that $F(x,t) \rightarrow F(x,t+1)$.*

B. Existing Evolutionary Approaches for Solving DMOPs

In the literature, existing evolutionary approaches for solving DMOPs are generally categorized into diversity-based methods, memory-based methods, and learning/prediction-based methods [7]. In particular, diversity-based methods

address dynamic changes by either increasing or sustaining population diversity. These methods can enhance diversity by introducing randomly generated individuals, engaging in hyper-mutation of specific existing solutions, or replacing solutions that rank low. Alternatively, they may focus on maintaining diversity throughout the evolutionary search process. Specific examples in this category include the steady-state and generational evolutionary algorithms discussed in [21], the evolutionary algorithm with diversity maintenance detailed in [22], and the co-evolution with diversity enhancement algorithm described in [23].

Moreover, memory-based methods often retain superior solutions from previous environments in a memory archive. The stored solutions are then utilized to guide the evolutionary search by generating initial individuals when the environment changes. For instance, Azzouz et al. proposed an evolutionary algorithm with a population management strategy that adaptively adjusts the contribution of memory, random search, and local search based on change severity [6]. Chen et al. [14] proposed a two-archive dynamic evolutionary algorithm with two co-evolving archives focusing on convergence and diversity, which complement each other through a mating selection mechanism. Similarly, Sahmoud and Topcuoglu [24] proposed a memory-based MNSGA-II algorithm, developing an explicit memory to store and reuse non-dominated solutions when similar environments reappear.

Algorithm 1 Typical procedure of learning and prediction-based evolutionary approaches for solving DMOPs

1. Initialization

a. Search population initialization.

b. The training set and learning model initialization.

2. Conventional multi-objective optimization in the static environments

3. ***Dynamic detection***

a. Re-evaluate some individuals or the whole population.

b. Evaluate the characteristics of the change.

4. ***New solutions prediction as dynamic response*****(If an environmental change is detected in Step 3)**

a. Configure the acquired knowledge as inputs to the learning model.

b. Employ the learning model to estimate the patterns of change.

c. Predict new initial population that optimally aligns with the estimated patterns.

5. Return to Step 2 and update the training set.

In contrast to the previous two types of algorithms, learning and prediction-based methods focus on extracting dynamic patterns and knowledge from past search experiences. These methods utilize the acquired knowledge to predict high-quality solutions, effectively guiding the search process during dynamic changes. Their superior adaptability and faster convergence make them particularly effective in rapidly changing environments. Consequently, learning and prediction-based methods have garnered significant attention and interest from researchers and practitioners in recent years [25], [26], [27]. Without loss of generality, the typical workflow of learning and prediction-based approaches can be outlined by Algorithm 1. This type of algorithm generally includes two key components, namely *Dynamic Detection* and *Population Prediction*, which are essential for effectively managing DMOPs. The process of *Population Prediction* begins by developing a learning model that captures dynamic patterns from previous search experiences. These patterns are subsequently utilized to estimate future changes and forecast initial solutions once a dynamic change has been detected.

In the literature, many learning and prediction-based methods have been proposed for solving DMOPs. Popular methods for constructing prediction models include the Kalman filter [28], denoising autoencoder [29], support vector regression (SVR) [30], Mahalanobis distance-based method [31], mixture-of-experts-based method [32], search guidance network [33], reinforcement learning [34] and feedback-based prediction method [35]. Recently, to achieve more accurate predictions in different dynamics, some approaches have focused on incorporating learning and knowledge transfer before solution prediction. For instance, Jiang et al. [36] designed an algorithmic structure based on transfer learning to address DMOPs. They employed transfer component analysis to overcome the limitations of independent and identical distributions. Additionally, they used a memory mechanism [37] to reserve the first-rank solutions from historical data, combined with a manifold transfer approach to handle DMOPs, addressing the bottleneck of slow convergence in transfer learning-based frameworks. In [30], on the basic of the decomposed multi-objective evolutionary algorithm, the paternal solution of each solution is determined to obtain sequential individuals, and the variable value of each individual in the next environment is fitted by an SVR approach. More recently, Hu et al. [38] also combined hybrid search strategies in a change response framework to discover promising regions with adjustable searching range controlled by the learned knowledge. To mitigate negative transfer, Lin et al. [39] proposed a discriminative predictor that minimizes feature and distribution divergences across different environments. This predictor identifies high-quality solutions from a large set of randomly generated candidates for newly encountered environments. Furthermore, Zhou et al. [40] developed a kernelized autoencoding method to facilitate knowledge transfer in evolutionary searches across multiple views. Chen et al. [20] also proposed a dynamic constrained optimization algorithm with population classification and selection to address drastic changes caused by constraints under diverse environments.

These aforementioned evolutionary approaches for solving DMOPs are rarely applied in isolation. Instead, cooperatively hybridizing multiple strategies often proves to be more effective [7], [41]. For instance, Feng et al. [29] proposed a hybrid method to utilize historical information by combining efficient prediction of DPOS with a memory pool to preserve high-quality solutions found along the evolutionary search.

However, as previously underlined, existing learning and prediction-based approaches primarily focus on knowledge learning within a single DMOP. This leaves the potential for learning generalized knowledge across multiple DMOPs

unexplored. This paper thus aims to fill this gap, and the details of the proposed approaches will be presented in the following Section III.

III. Proposed Method

This section provides an in-depth explanation of the proposed Centralized Prediction-based Dynamic Multi-Objective Evolutionary Algorithm, abbreviated by CP-DMOEA. Figure 1 depicts the complete framework of the CP-DMOEA. The process starts with initializing the input dynamic multi-objective problem (DMOP). Subsequently, a conventional multi-objective optimization evolutionary algorithm (MOEA) is employed as the primary solver. The evolutionary search process then proceeds in a routine manner, adhering to the standard procedures of static optimization. In contrast, additional dynamic change detectors are placed within the iterative search process to trigger the *Centralized Prediction* stage. In this stage, a prediction model is pre-trained using the non-dominated solutions collected from different solved DMOP tasks, as well as the extended *Dynamic Features*. This model is constructed to predict the re-initialization population for the subsequent search of the ongoing task that has been detected as changed. The search process continues iteratively until the predefined termination criteria are satisfied.

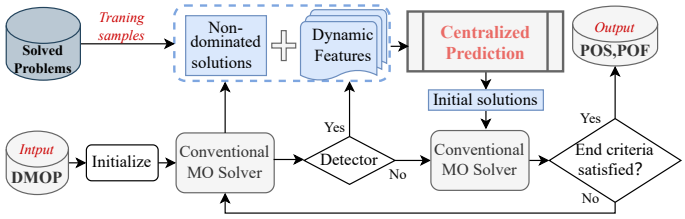


Fig. 1. Outline of the proposed CP-DMOEA framework.

Algorithm 2 presents the overall framework of the proposed CP-DMOEA in solving DMOPs. Specifically, randomly generated solutions are used for initialization (line 1). When an environmental change is detected (see Section III-A) and the change index is equal to one, random solutions with size N_P are used to form the first initial population (line 5). Centralized learning-based prediction is activated for change response when at least two environmental changes are detected. CP-DMOEA first conducts the supervised learning process to update the pre-trained centralized learning model $\mathcal{M}$ with detected dynamic features and labels via Algorithm 3 (line 7). Then the initial population generator is performed to predict the initial population IP to restart the search via Algorithm 4 (line 8). If no environmental change is detected, a basic multi-objective optimization solver is used for static evolutionary search (line 12). In what follows, the details of the dynamic detector, the centralized prediction model, and the initial population generator are successively presented from section III-A to section III-C, respectively.

A. Dynamic Detector

Dynamic environmental detection provides essential prior knowledge for the prediction of change patterns [42], [43].

Algorithm 2 The overall framework of CP-DMOEA in solving DMOPs.

Input: $\{\mathcal{S}\}$: Training samples of solved DMOPs.
N_P: Population size
$F(\mathrm{x}, t)$: The dynamic optimization problem functions.
$MOEA(\cdot,\cdot)$: The basic multi-objective optimization algorithm.
Output: POS_t: Approximate DPOS over a time series.

1: Initialize a population of size N_P.
2: **while** End criteria is not satisfied **do**
3: **if** Environment change is detected **then**
4: **if** $t \leq 1$ **then**
5: $IP \leftarrow$ Randomly generated N_P solutions.
6: **else**
7: Train the centralized learning model $\mathcal{M}$ via Algorithm 3.
8: $IP \leftarrow$ Generate the initial population as change response via Algorithm 4.
9: **end if**
10: $t = t + 1$
11: **end if**
12: $POS_t = MOEA(IP, F(\mathrm{x}, t))$
13: **end while**
14: **return** POS_t

In the literature of DMO, existing dynamic detection approaches can be generally categorized into three categories, i.e., 1) random re-evaluation, 2) sensor-based detection and 3) population-based detection [7]. Random re-evaluation-based detection re-evaluates a proportion of the population at each generation randomly, which is the most commonly used technique for change detection in DMO. In contrast, sensor-based detection selects determined individuals as sensors and places them in the iterative process to detect environmental dynamics by monitoring their objective values or feasibility. In addition, population-based detectors rely on the fitness evaluations of the entire population to detect changes.

Understanding the features of dynamics can help the algorithm effectively adapt to changes. Taking this cue, a sensor-based detection method is proposed in CP-DMOEA to monitor the environmental changes and provide dynamic features (i.e., dynamic index and severity tag) for model learning in the *Centralized Prediction* stage. To assess the severity of the change, a random set with a fixed number of points is chosen to act as sensors. Note that the selected points are from outside the current population and may belong to any part of the fitness landscape. This selection strategy outperforms other from-population sensor-based alternatives, according to [44]. In particular, for each objective function, the differences between values evaluated before and after the dynamic occurs are calculated separately as follows:

$$SC_i = \sum_{j=1}^{s} \frac{f_{i,j}(t) - f_{i,j}(t-1)}{f_{i,j}(t) + c} \tag{5}$$

where s is the number of selected sensors and SC_i denotes the severity of change in the i-th objective function. The objective

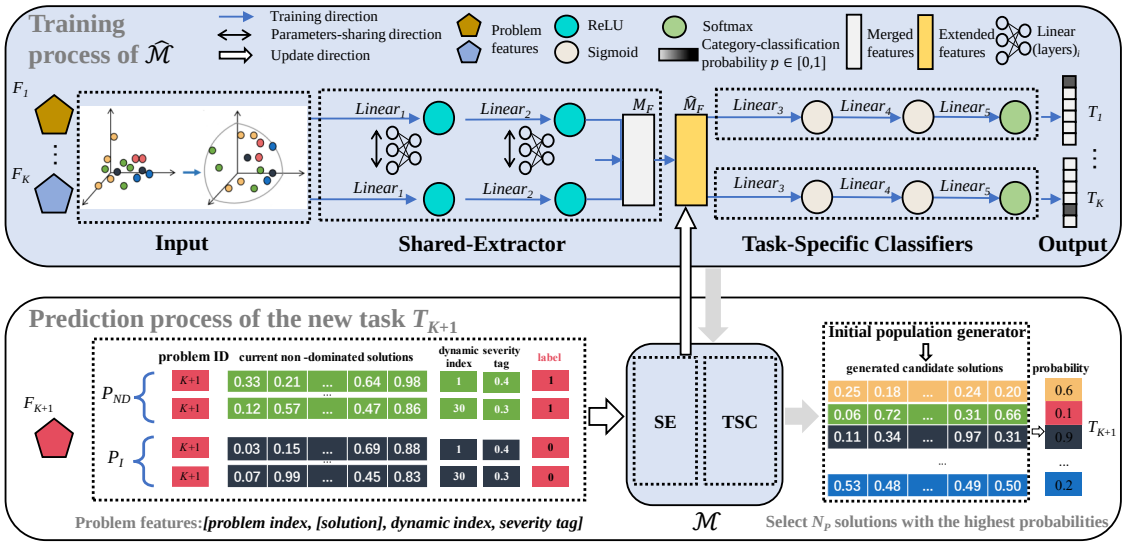


Fig. 2. Training and prediction processes of the centralized learning model.

values of j-th sensor before and after the change are termed as $f_{i,j}(t)$ and $f_{i,j}(t-1)$, respectively. The term c is a constant value added to ensure the denominator is not equal to zero. Then the detected change severity of the encountered DMOP can be defined as:

$$SC = (m-1) \times \max(SC_1, SC_2, \cdots, SC_m) \tag{6}$$

where the $(m-1)$ term is used to adapt the severity for m number of objectives. Thus, SC can be adopted as one of the most crucial features to represent the severity level of the detected change. In addition, the current dynamic index t is also counted by the detector and serves as another indispensable dynamic feature.

B. Centralized Learning-based Prediction

To uncover diverse dynamic patterns beneath various types of DMOPs and thus enable dynamic multi-objective optimization across different problems, a centralized learning-based population prediction model is proposed in this paper, which serves as the core dynamic response operator of CP-DMOEA. Specifically, the prediction model consists of two modules: 1) the shared extractor with two linear layers merging the diverse features of solved problems; and 2) the task-specific classifiers with three linear layers processing the combined dynamic features and classifying the type of the current change. The classifiers output predicted values of the probability that the candidate solution matches the current task, enabling the initial population generator to predict the entire population efficiently for locating new optima in the changed environment. The following sections will detail the model's training process and the structure of each module.

1) Centralized learning process: To prepare the input training samples for the pre-trained centralized learning model, the non-dominated solutions obtained by the basic DMOP optimizer[1] on the solved DMOPs are collected as *positive* samples, while the re-initialized solutions at the beginning of different change steps are gathered and labeled as *negative* samples. Note that the adopted solved problems are picked from the DMOPs with different types of dynamics (i.e., different dynamic frequencies and severities). In centralized learning, the model is provided with the solutions of solved problems (the searched non-dominated solutions are *positive* samples and the randomly re-initialized solutions are

Algorithm 3 Training process of the centralized learning model.

Input: K: Number of input solved DMOPs.
N_s: Maximum number of samples.
$\{\mathcal{F}_r\}_{r=1}^{K}$: Input solution features of solved DMOPs for pre-training.
$\{\mathcal{Y}_r\}_{r=1}^{K}$: Input solution labels.
P_I: Re-initialized population at the beginning of the current step.
P_{ND}: Current non-dominated solution set.
Output: $\mathcal{M}$: Trained centralized learning model.

```
for r ← 1 to K do
    Obtain F̂_r and Ŷ_r by randomly select N_r^t samples from F_r and Y_r, respectively.
    Train M̂ with F̂_r and Ŷ_r.
end for
while Update condition is met do
    Extend P_I and P_ND with detected dynamic features and labels to obtain F̂_{K+1} and Ŷ_{K+1}, respectively.
    Update M̂ with F̂_{K+1} and Ŷ_{K+1}.
end while
M ← M̂
return M
```

[1]A simple change response strategy that introduces 30% randomly generated solutions into the population and hyper-mutates the other 30% existing solutions [45].

negative samples) associated with the dynamic features, i.e., $[problem\_index, [solution], dynamic\_index, severity\_tag]$, where the $dynamic\_index$ corresponds to the index of the change, $severity\_tag$ is assigned by the SC calculated via Equation 6. The model can learn the inherent and general correlations between the features of a solution and its quality label. Each solution is assigned a label that represents its selection probability: a positive input sample gets a value of 1, while a negative sample is given a value of 0. To be more specific, Fig. 2 gives an example of obtaining the labels of the candidate solutions. Since the DMOPs used for pre-training may not possess the dynamic patterns exhibited in the DMOP being currently addressed, we propose to update the model online using supervised data obtained from the current population when a dynamic is detected.

The pseudocode of the training process for the centralized learning model is elucidated in Algorithm 3. Particularly, each row of the matrix $\mathcal{F}$ denotes the dynamic features of a solution. The pre-trained centralized model $\hat{\mathcal{M}}$ is trained with $N_r^t (N_r^t \leq N_s)$ randomly selected samples with the corresponding known solution labels (see lines 1-4). When a new task arrives with a detected change, the trained $\hat{\mathcal{M}}$ is updated with newly detected dynamic features and labels as supervised learning (see lines 5-8). The training process will continue until a DMOP optimization task terminates, resulting in a centralized model that can realize optimization knowledge learning across problems.

2) Shared extractor: The centralized learning model is trained by sequentially inputting the features from each solved DMOP in each round. As shown in Fig. 2, the shared extractor, which is based on an artificial neural network, consists of two linear layers with rectified linear unit activation functions [46]. In the training phase, input features are first transformed into hidden representations through these linear layers, utilizing the weights of the network. This transformation process is described by the following formulation:

$$h(x)_i = \sigma\{\sum_{i=1}^{l}\sum_{j=1}^{d}(w_{ij}x_j + b_i)\} \tag{7}$$

where h_i denotes the mapping in the i-th layer of the shared-extractor, while l is the number of neurons in the hidden layers, and d is the dimension of the input vector. The proposed shared extractor undergoes continuous updates and fine-tuning during each training round. This process utilizes the features from the current DMOP task that requires resolution. Generally, the shared feature extractor fuses features by utilizing shared weights during model training. Benefiting from this, the encountered task can leverage the parameters from previously solved tasks in the shared extractor and proceed to train both the extractor and the classifier.

3) Task-specific classifier: The task-specific classifiers are designed to work in conjunction with the shared extractor for their respective tasks. This setup enables the classifiers to utilize the integrated features and produce classification probabilities for new initial solutions, accommodating various types of dynamics. As depicted in Figure 2, each classifier processes the features provided by the shared extractor through a three-layer neural network. Specifically, the first two layers use the Sigmoid activation function to normalize the feature values. For the final classification, the Softmax function is applied to convert the output values into probabilities within the $[0, 1]$ interval. Last but not least, the cross-entropy loss function with the Adam optimizer [47] is adopted to optimize the model and tune the parameters.

Algorithm 4 Initial population generator.

Input: P_{ND}: Current non-dominated solution set.
N_P: Population size
k_s: Multiple of the number of candidate solutions over N_P.
$\mathcal{M}$: Centralized learning model trained via Algorithm 3.
Output: IP: Initial population.

1: Randomly divide P_{ND} into two subsets, each consists of half non-dominated solutions, denoted as P_{ND}^G and P_{ND}^L, respectively.
2: $C \leftarrow \{\}$
3: **for** $i \leftarrow 1$ to k_s **do**
4: **for** $j \leftarrow 1$ to $N_P/2$ **do**
5: $C_i^G \leftarrow [C_i^G, P_{ND,j}^G + n_{i,j}^{normal}]$
6: $C_i^L \leftarrow [C_i^L, P_{ND,j}^L + n_{i,j}^{laplace}]$
7: **end for**
8: $C \leftarrow [C, [C_i^G, C_i^L]$
9: **end for**
10: Obtain selection possibility of each item in C using $\mathcal{M}$.
11: Sort C according to the estimated selection possibility concerning the current DMOP task.
12: $IP = C[1 : N_P]$
13: **return** IP

C. Initial Population Generator

Change response is the most crucial stage of algorithmic design where dynamic multi-objective optimization algorithms react to environmental changes that have been detected [7]. Since DMOPs tend to have short stationary periods, it's important for dynamic optimization algorithms to quickly respond to these changes by creating an initial population that is suited to the current problem-solving environment.

The centralized learning model $\mathcal{M}$, trained with the dynamic features across problems (refer to Algorithm 3), can provide the classification probability to determine whether an initial solution is "qualified" for the current environment after the change. Based on this, the initial population generator needs to provide a series of new solutions for the $\mathcal{M}$ to identify and form the initial population. In particular, as referred to Algorithm 4, k_s times the population size of candidate solutions are randomly generated by perturbing current non-dominated solutions obtained before the change, with a combination of Gaussian and Laplace-distributed noise (see lines 3-9 in Algorithm 4). The scale of both noises is controlled by the detected $severity\_tag$. Then, each of the $k_s \times N_P$ number of newly generated candidate solutions is assessed by centralized learning model $\mathcal{M}$ with a probability vector indicating the possibility the solution corresponds to

different DMOP tasks. Finally, the N_P number of candidate solutions with the highest estimated probabilities are selected to constitute the initial population for guiding the evolutionary search toward the changed optima.

D. Complexity Analysis

The computational cost of the prediction component comes from the training of the centralized learning model, whose time complexity is determined by the number of samples and the size of the weight matrix. Since the weight dimension of the model is set much smaller than the sample size used for training (specific settings can be referred to Section IV-C), the complexity of the proposed prediction model is $\mathcal{O}(N_s)$. The majority of the existing learning and prediction-based approaches have a time complexity of $\mathcal{O}({N_P}^2)$ [7]. Therefore, under commonly used experimental settings for solving DMOPs, the time complexity of CP-DMOEA is comparable to existing methods and does not impose a computational burden on dynamic optimization. This is verified by analyzing the specific algorithm runtime in Section V-B.

IV. Experimental Design

A. Test Instances

To investigate the performances of the proposed centralized prediction-based dynamic multi-objective optimization method with learning across problems, namely CP-DMOEA, against the state-of-the-art learning and prediction-based methods, the widely adopted DMOP benchmark suite for IEEE CEC2018 Competition on DMO (also known as DF benchmarks) [48] is adopted to conduct comparative experiments in this paper. The DF test suite comprises fourteen benchmark DMOPs, including nine two-objective and five three-objective functions, each featuring a range of dynamic characteristics. These dynamics include time-varying DPOF and DPOS, fluctuating objective function landscapes, evolving variable linkages, and mixed time-varying convexity and concavity. The dynamics of the DMOP benchmark functions are controlled by a discretized time variable t, which is expressed as:

$$t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor, \tag{8}$$

where τ herein represents the generation counter and $\lfloor \cdot \rfloor$ is the floor operator. According to the frequency and severity of dynamic defined in Section II-A, a larger value of τ_t leads to less frequent dynamics of the given DMOP, while a larger value of n_t results in slighter changes, respectively.

In the experiments, according to [40], [48], 5 fixed dynamic configurations and 2 modified shifty settings of dynamic changes were investigated for the comprehensive assessment. Table I outlines the total 7 configurations to form two groups of comparative experiments. In particular, the C_1 and C_2 configurations are used to test the robustness of the change response strategy, with C_1 reflecting frequent dynamics and C_2 representing severe dynamics. Meanwhile, the C_3, C_4, and C_5 configurations are designed to assess algorithm performance across DMOPs with varying degrees of dynamic severity. To complete the training of the centralized prediction model, we randomly select DMOP instances under two different dynamic settings in Group G_1 and collect the non-dominated optimized solutions obtained by a basic DMOP optimizer. These positive samples are associated with the re-initialized solutions at the beginning of different change steps, to form the training input of the pre-trained centralized learning model. Subsequently, the remaining three dynamic settings in Group G_1 and two shifty dynamic settings in Group G_2 will be used to evaluate the performance of the algorithms in handling frequent, severe, and unpredictable dynamics, until all settings have been tested. For example, if DF1-C1 and DF1-C2 are selected as training instances, then DF1-C3 to DF1-C7 from both groups will be used as test instances. This is to construct similar optimization tasks for positive cross-problem knowledge transfer. Note that the group experiments above are independent of each other and the model is trained independently in each round.

TABLE I
Fixed and Shifty Dynamic Configurations of the DMOPs

$Group$	$Configuration$	n_t	τ_t
G_1	C_1	10	10
	C_2	1	30
	C_3	5	10
	C_4	2.5	10
	C_5	1	10
G_2	C_6	1, 5, 10	10
	C_7	5	10, 30

B. Performance Metrics

In the literature, quite a few performance indicators have been proposed to quantify the ability of dynamic multi-objective optimization algorithms in solving DMOPs [17], [21], [36], [49]. Two common groups of performance metrics are used to evaluate the performance of the algorithms being compared, focusing on convergence and diversity, respectively. The first group includes the inverted generational distance (IGD) [50], along with its two dynamic variants: mean IGD (MIGD) [28], [37] and dynamic mean IGD (DMIGD) [36]. The second group comprises the Hypervolume (HV) [51], which also has its variants: mean HV (MHV) [52] and dynamic mean HV (DMHV). To be more specific, the IGD metric is formulated as:

$$\text{IGD} = \frac{\sum_{v^* \in POF^{t*}} \min_{v \in POF^t} \parallel v^* - v \parallel}{\mid POF^{t*} \mid} \tag{9}$$

where POF^t is the current approximated POF searched by the algorithm, and POF^{t*} represents the true POF at the current time step t. v and v^* denote the objective values of the individuals in POF^t and POF^{t*}, respectively. $\parallel v^* - v \parallel$ calculates the Euclidean distance between v^* and v. The MIGD metric is a dynamic variant of IGD that computes the means of the IGD values obtained over a certain number of periods, which is defined as:

$$\text{MIGD} = \frac{\sum_{t \in T} \text{IGD}(POF^{t*}, POF^t)}{|T|} \tag{10}$$

where T represents a set of discrete time periods, with $|T|$ denoting the cardinality of T. The DMIGD metric is used to

show the robustness of the algorithms in solving DMOPs with different dynamic configurations, which is defined as:

$$\text{DMIGD} = \frac{\sum_{C_i \in E} \text{MIGD}(C_i)}{|E|} \tag{11}$$

where E is the set of different dynamic characteristics and C_i denotes a specific configuration of the dynamic environment. In the experiments, as two groups of 7 dynamic configurations are set for each DMOP (see Table I), $|E|$ is equal to 7. The HV metric calculates the hypervolume of the region dominated by the searched solutions, which is defined as:

$$\text{HV} = \Lambda(\bigcup_{v \in POF} v' | ref \succ v' \succ v) \tag{12}$$

where Λ is the Lebesgue measure and $ref \in \mathbb{R}^m$ is the reference point. The MHV and DMHV metrics are two dynamic variants of HV, which are consistent with the modification to MIGD and DMIGD, respectively.

C. Compared Algorithms and Parameter Settings

The proposed CP-DMOEA is evaluated against five advanced learning and prediction-based approaches to validate the dynamic response performances. Specifically, the algorithms used for comparison in the experiments are briefly described as follows:

- *SVR-MOEA/D* [30]: A dynamic DMO predictor based on the MOEA/D-DE solver determines the paternal solution for each solution to generate sequential individuals, with the variable values of each individual in the next environment being predicted using SVR technique.
- *Tr-MOEA/D* [36]: A transfer learning-based algorithmic framework predicts the initial population by utilizing transfer component analysis for knowledge transfer to overcome the limitations of independent and identical distributions.
- *AE-MOEA/D* [29]: A predictor based on knowledge learning and transfer, which employs the denoising autoencoder to learn the correlations between populations in two adjacent environments and transfer the previous high-quality search experience to the new environment.
- *MV-MOEA/D* [40]: A multi-view learning approach utilizing a kernelized autoencoding model to make predictions separately from both the decision space and the objective space.
- *MSTL-MOEA/D* [52]: A multi-source transfer learning method utilizes clustering-based manifold transfer learning and multi-source TrAdaboost to leverage historical information from all previous dynamics in constructing a prediction model.

For the fair comparisons, following the basic optimization solver used in the compared algorithms, our proposed centralized learning-based prediction model also takes MOEA/D [53] as the static baseline (labeled as *CP-MOEA/D*). In the comparisons, other general parameters of experiments are set as follows. All algorithms use the random selection mechanism of sensor-based change detection to detect dynamics (see Section III-A). 10% of individual solutions from the population are randomly selected as detectors and their objective values are re-evaluated at the beginning of each iteration. If the average of the absolute values of the difference SC is greater than $1e-5$, it indicates a detected dynamic in the environment. In addition, according to [48], the environment for each problem instance changes 30 times. The population size N_P for all algorithms is set to 100, and the variable dimension d for all problems is set to 10. Each algorithm independently runs 20 times. The specific parameter settings of the centralized learning-based prediction model in *CP-MOEA/D* are listed as follows[2]:

- Maximum number of samples for model training: $N_s = N_P * 30 * 2 * 2 = 12000$. (According to [48], 30 is the maximum number of dynamics. The last two multipliers 2 mean positive & negative sample sets and two randomly selected training cases, respectively.)
- The six-layer neural network-based centralized learning model[3] was trained for 50 rounds with the batch size and the learning rate set to 32 and $1e-4$, respectively. The weight dimensions of the linear layers in the task-specific classifier are 60, 30, and 6, respectively [54]. The algorithm is repeated 20 times with different random seeds, the mean and standard deviation of the results are reported in the tables.
- The dimension d_{in} of the input vectors (i.e., $[problem\_index, [solution], dynamic\_index, severity\_tag]$) for the prediction model is set to $d_{in} = 13$. k_s is set to 4 and the dimension of the output vectors d_{out} is equal to $k_s \times N_P = 400$.

The other evolutionary parameters for the algorithms based on the MOEA/D optimizer are maintained in the same settings as those adopted in references [30], [36] and [40], respectively.

V. Experimental Results and Analysis

This section presents detailed empirical studies to assess the performance of the proposed centralized prediction with learning across problems to solve DMOPs. The evaluation is performed on the widely adopted DMOP benchmarks and compared against five learning and prediction-based methods. The experimental results obtained on the DF benchmarks with the conventional fixed dynamic settings and our proposed modification of shifty dynamic settings are summarized and analyzed. Initially, the solution quality is evaluated using two groups of numerical metrics, which are presented in tabular results and summarized by algorithm rankings. This is followed by a further analysis of the convergence performance and the cross-problem knowledge transfer.

A. Numerical Performance Comparison of Solution Quality

The averaged MIGD and MHV statistical results with the standard deviations obtained by *CP-MOEA/D* and five compared algorithms under 5 fixed dynamic configurations (refer

[2]To ensure a fair comparison, the parameter settings for model training are not specifically tuned with additional validation datasets for better performance. These settings are based on the configurations for training a general neural network [54], [55].

[3]The study on the impact of the number of network layers on prediction performance can be referred to in the supplementary materials.

TABLE II
MEAN AND STANDARD DEVIATIONS OF MIGD VALUES OBTAINED BY 6 ALGORITHMS WITH 5 FIXED DYNAMIC SETTINGS

Problems	(n_t,τ_t)	*SVR-MOEA/D*	*Tr-MOEA/D*	*AE-MOEA/D*	*MV-MOEA/D*	*MSTL-MOEA/D*	*CP-MOEA/D*
DF1	(10,10)	9.9427E-02 (1.3937E-02)+	1.7030E-02 (9.1034E-03)+	5.8867E-02 (4.5844E-02)+	1.4022E-02 (4.3166E-03)≈	8.3313E-02 (8.2516E-03)+	**1.2513E-02 (6.6326E-03)**
	(1,30)	**5.5042E-03 (2.4685E-03)–**	1.7175E-02 (6.0347E-03)+	1.3062E-01 (6.1357E-02)+	1.7790E-02 (7.4966E-03)+	6.6350E-02 (9.4792E-03)+	9.3671E-03 (4.0526E-03)
	(5,10)	1.3452E-01 (2.5226E-02)+	1.9955E-02 (1.9053E-03)≈	7.6897E-02 (4.7661E-02)+	2.5200E-02 (1.9123E-02)+	8.2915E-02 (9.7106E-03)+	**1.9152E-02 (8.6764E-03)**
	(2.5,10)	2.6566E-01 (3.1520E-02)+	4.5946E-02 (2.5566E-02)+	2.3178E-01 (7.9999E-02)+	4.2477E-02 (2.1113E-02)≈	1.1912E-01 (1.5158E-02)+	**3.6280E-02 (1.3465E-02)**
	(1,10)	2.3835E-01 (4.4688E-02)+	5.8406E-02 (1.3437E-02)≈	9.3783E-01 (8.3711E-02)+	**5.1142E-02 (1.0142E-02)**≈	1.6332E-01 (1.6308E-02)+	6.7543E-02 (2.0472E-02)
DF2	(10,10)	5.4687E-02 (9.9333E-03)+	1.8202E-02 (1.0415E-02)≈	6.2012E-02 (2.3732E-02)+	2.8852E-02 (2.5036E-02)+	3.7533E-02 (1.3560E-02)+	**1.7159E-02 (1.1702E-02)**
	(1,30)	**5.0367E-03 (2.3030E-04)–**	1.1558E-02 (4.2142E-03)+	4.3379E-02 (3.6074E-02)+	1.4285E-02 (7.4865E-03)+	1.7318E-02 (5.4003E-03)+	9.6084E-03 (6.5735E-03)
	(5,10)	8.9449E-02 (1.5467E-02)+	2.3551E-02 (1.0912E-02)≈	9.4036E-02 (3.0313E-02)+	2.7605E-02 (1.9414E-02)≈	4.0502E-02 (1.0376E-02)+	**2.2994E-02 (2.0114E-02)**
	(2.5,10)	1.4802E-01 (1.7951E-02)+	6.4191E-02 (7.4648E-02)+	1.4486E-01 (4.3120E-02)+	**2.3145E-02 (7.0737E-03)**≈	6.3408E-02 (8.2065E-03)+	2.4385E-02 (2.8243E-02)
	(1,10)	1.9155E-01 (3.7265E-02)+	5.9982E-02 (2.9287E-02)≈	6.8955E-01 (8.1674E-02)+	5.8219E-02 (4.5801E-02)≈	1.3385E-01 (3.1337E-02)+	**5.7396E-02 (5.0786E-02)**
DF3	(10,10)	4.4562E-01 (3.0149E-02)+	2.7494E-01 (1.3271E-02)+	1.5777E-01 (7.6047E-02)≈	2.2643E-01 (1.3125E-01)+	**1.3635E-01 (1.6836E-02)**≈	1.3983E-01 (6.3279E-02)
	(1,30)	3.3456E-01 (2.4690E-02)+	1.2014E-01 (1.8033E-02)≈	3.3179E-01 (2.3459E-02)+	**1.1612E-01 (7.3164E-02)**≈	1.4212E-01 (1.6924E-02)≈	1.3621E-01 (4.7010E-02)
	(5,10)	4.3026E-01 (2.4095E-02)+	2.6010E-01 (1.8158E-02)+	2.1963E-01 (9.1768E-02)≈	2.4959E-01 (1.0154E-01)+	2.0509E-01 (1.7798E-02)≈	**2.0362E-01 (7.2062E-02)**
	(2.5,10)	3.8812E-01 (3.8774E-02)+	2.1170E-01 (1.8949E-02)+	3.6561E-01 (5.3980E-02)+	1.7948E-01 (8.3996E-02)+	2.1796E-01 (1.9940E-02)+	**1.3897E-01 (4.3389E-02)**
	(1,10)	5.0384E-01 (2.0947E-02)+	1.5329E-01 (2.2963E-02)+	4.5992E-01 (2.1624E-02)+	1.4426E-01 (4.3663E-02)≈	2.7913E-01 (4.1785E-02)+	**1.3593E-01 (9.9654E-03)**
DF4	(10,10)	9.1818E-02 (1.0187E-03)+	1.1772E-01 (1.5715E-02)+	1.2366E-01 (2.4296E-02)+	1.2955E-01 (2.1407E-02)+	4.9394E-01 (2.0565E-02)+	**4.3657E-02 (3.6994E-02)**
	(1,30)	2.1998E-02 (2.7209E-04)≈	4.2462E-02 (1.2128E-02)+	4.1377E-02 (1.0288E-02)+	3.9224E-02 (6.9094E-03)+	5.9206E-01 (5.3746E-02)+	**2.1684E-02 (8.3373E-03)**
	(5,10)	1.2153E-01 (1.9072E-02)≈	1.2311E-01 (2.4970E-02)≈	1.2663E-01 (2.7586E-02)≈	**1.1915E-01 (2.4585E-02)**≈	5.4811E-02 (4.2102E-02)+	1.2122E-01 (1.4017E-02)
	(2.5,10)	**9.1849E-02 (9.6186E-03)**≈	1.1015E-01 (1.9350E-02)+	9.9298E-02 (1.1774E-02)≈	1.0287E-01 (1.8881E-02)≈	4.7712E-01 (2.7691E-02)+	9.8063E-02 (1.6140E-02)
	(1,10)	1.1978E-01 (2.7982E-02)+	6.1115E-02 (9.4059E-03)+	9.3864E-02 (3.0596E-02)+	4.9601E-02 (1.6363E-02)+	5.9240E-01 (3.2331E-02)+	**3.7518E-02 (1.1537E-02)**
DF5	(10,10)	1.1786E-02 (9.2051E-04)+	1.4513E-02 (4.2597E-03)+	1.2371E-02 (4.1958E-03)+	1.0439E-02 (1.1439E-03)≈	3.4612E-02 (1.5763E-03)+	**9.2336E-03 (1.6797E-03)**
	(1,30)	**4.6829E-03 (3.0923E-05)**≈	1.0129E-02 (1.3972E-03)+	2.2424E-02 (2.6797E-02)+	9.4244E-03 (1.1755E-03)+	2.2652E-02 (1.5015E-03)+	5.8514E-03 (5.0317E-04)
	(5,10)	4.1078E-02 (2.1489E-02)+	1.6426E-02 (8.6040E-03)+	1.8029E-02 (1.6554E-02)+	1.1183E-02 (2.1171E-03)≈	3.6684E-02 (2.0334E-03)+	**9.9913E-03 (1.1413E-03)**
	(2.5,10)	2.0349E-01 (4.1019E-02)+	1.3889E-02 (1.5160E-03)+	3.4879E-02 (1.0731E-02)+	1.2256E-02 (1.7598E-03)≈	5.0774E-02 (1.8099E-02)+	**1.0948E-02 (1.3250E-02)**
	(1,10)	1.9824E-02 (6.2557E-03)+	3.8425E-02 (1.0211E-02)+	2.9461E-01 (1.2138E-01)+	1.6689E-02 (5.2777E-03)≈	1.4100E-01 (3.4818E-02)+	**1.4067E-02 (8.5717E-03)**
DF6	(10,10)	3.4651E+00 (5.8456E-01)+	1.0713E+00 (2.8422E-01)+	3.3310E+00 (1.8680E+00)+	5.6021E-01 (2.0117E-01)+	8.6584E-01 (1.2306E-01)+	**2.1934E-01 (1.9109E-01)**
	(1,30)	5.5959E-01 (7.0076E-02)+	3.8446E-01 (2.7810E-02)+	5.7520E-01 (8.1478E-02)+	1.5506E-01 (2.3250E-02)≈	1.9740E-01 (2.1482E-02)+	**1.3671E-01 (4.0859E-02)**
	(5,10)	4.3968E+00 (7.0642E-01)+	1.2814E+00 (3.9050E-01)+	3.3113E+00 (1.5408E+00)+	9.2799E-01 (4.8344E-01)≈	**5.4628E-01 (2.0216E-01)–**	7.9330E-01 (4.6224E-01)
	(2.5,10)	1.1746E+00 (2.5784E-01)+	5.7628E-01 (5.3989E-02)+	1.0472E+00 (5.1391E-01)+	2.8885E-01 (6.3289E-02)+	8.0181E-01 (2.1237E-01)+	**2.0365E-01 (5.7534E-02)**
	(1,10)	6.4976E-01 (9.4893E-02)+	4.4211E-01 (2.2433E-02)+	5.8973E-01 (7.9075E-02)+	1.6992E-01 (3.1595E-02)+	5.1902E-01 (1.8860E-01)+	**9.8434E-02 (2.7422E-02)**
DF7	(10,10)	3.8059E-01 (3.3149E-02)+	2.2296E-01 (5.6945E-02)+	3.1206E-01 (1.0496E-01)+	**1.7086E-01 (4.2469E-02)–**	8.8284E-01 (4.3874E-02)+	2.0776E-01 (5.4016E-02)
	(1,30)	**1.2778E+00 (7.0306E-02)–**	1.7190E+00 (4.9491E-01)+	2.5654E+00 (2.6392E-01)+	2.5919E+00 (3.6802E-01)+	4.3674E+00 (3.6405E-01)+	1.5260E+00 (3.3906E-01)
	(5,10)	4.7707E-01 (3.7654E-02)+	2.6546E-01 (4.1794E-02)+	4.6314E-01 (1.5170E-01)+	3.5346E-01 (1.4500E-01)+	1.1460E+00 (1.7744E-01)+	**1.7931E-01 (4.2523E-02)**
	(2.5,10)	8.5148E-01 (6.3386E-02)+	7.2006E-01 (9.5435E-02)≈	9.9311E-01 (7.7662E-02)+	1.0969E+00 (1.2576E-01)+	1.7390E+00 (1.7950E-01)+	**6.9328E-01 (4.6925E-02)**
	(1,10)	**1.5673E+00 (1.2124E-01)–**	1.8238E+00 (3.2338E-01)≈	2.7774E+00 (2.0367E-01)+	2.8401E+00 (1.4943E-01)+	3.7851E+00 (2.8663E-01)+	1.8365E+00 (2.0969E-01)
DF8	(10,10)	8.3044E-02 (7.5518E-03)+	9.3185E-02 (1.5418E-02)+	3.7937E-01 (9.8076E-02)+	1.7846E-01 (4.7569E-02)+	1.7091E-01 (3.2501E-02)+	**6.3724E-02 (9.4605E-03)**
	(1,30)	5.3549E-02 (1.4882E-03)≈	8.2077E-02 (5.2186E-02)+	2.1928E-01 (5.7026E-02)+	1.6202E-01 (4.4098E-02)+	1.3845E-01 (3.0316E-02)+	**5.2990E-02 (5.3380E-03)**
	(5,10)	**8.1660E-02 (2.3410E-03)–**	1.0486E-01 (1.6498E-02)≈	2.8597E-01 (9.0153E-02)+	2.0934E-01 (3.4481E-02)+	1.6417E-01 (3.1825E-02)+	1.3789E-01 (3.0286E-02)
	(2.5,10)	8.0325E-02 (2.0085E-03)≈	1.2650E-01 (4.2268E-02)+	3.1136E-01 (8.4715E-02)+	2.0742E-01 (3.4597E-02)+	1.4947E-01 (3.1473E-02)+	**7.7954E-02 (2.6315E-03)**
	(1,10)	**7.3980E-02 (1.2718E-02)**≈	1.0764E-01 (3.7231E-02)+	2.1145E-01 (7.6802E-02)+	1.9932E-01 (6.0840E-02)+	1.4168E-01 (3.5430E-02)+	7.4537E-02 (1.0406E-02)
DF9	(10,10)	2.8293E-01 (5.7400E-02)+	6.6019E-02 (1.6359E-02)+	4.5146E-01 (1.1172E-01)+	3.1942E-02 (4.5129E-03)+	1.2333E-01 (1.0516E-02)+	**2.1827E-02 (3.3736E-03)**
	(1,30)	1.2202E-01 (3.5805E-02)+	4.5487E-02 (1.8535E-02)+	4.0853E-01 (9.6714E-02)+	2.3758E-02 (2.7295E-03)+	1.4568E-01 (2.1447E-02)+	**1.4360E-01 (2.9424E-03)**
	(5,10)	3.0401E-01 (6.6189E-02)+	6.5768E-02 (9.3095E-03)+	6.7472E-01 (1.4394E-01)+	**3.6492E-02 (8.3702E-03)**≈	1.6154E-01 (2.8654E-02)+	4.2518E-02 (9.4367E-03)
	(2.5,10)	1.9933E-01 (5.3866E-02)+	6.1455E-02 (8.3039E-03)+	1.1068E+00 (1.0039E-01)+	3.7516E-02 (1.0721E-02)+	2.4863E-01 (2.2384E-02)+	**2.5838E-02 (1.3514E-02)**
	(1,10)	3.5209E-01 (7.7821E-02)+	8.2888E-02 (2.9222E-02)+	1.2102E+00 (2.2979E-01)+	3.5934E-02 (6.2848E-03)≈	5.4606E-01 (4.3749E-02)+	**2.8907E-02 (1.5705E-02)**
DF10	(10,10)	2.4643E-01 (1.1387E-02)+	1.6463E-01 (1.9533E-02)+	3.3808E-01 (1.0343E-01)+	1.4887E-01 (1.6076E-02)+	4.1961E-01 (1.9384E-02)+	**1.3271E-01 (9.9894E-03)**
	(1,30)	1.7349E-01 (1.4928E-02)+	2.0710E-01 (4.4932E-02)+	1.6361E-01 (5.8823E-02)+	1.3118E-01 (1.6025E-02)+	3.2769E-01 (1.6555E-02)+	**1.1963E-01 (1.1640E-01)**
	(5,10)	2.2395E-01 (1.0434E-02)+	1.3621E-01 (8.1155E-03)≈	2.0224E-01 (9.3603E-02)+	**1.3604E-01 (1.6769E-02)**≈	3.4831E-01 (1.6036E-02)+	1.4226E-01 (1.1271E-02)
	(2.5,10)	2.2676E-01 (2.5067E-02)+	1.6812E-01 (2.0096E-02)+	2.0527E-01 (8.8628E-02)+	1.5611E-01 (1.5402E-02)+	3.6445E-01 (1.7214E-02)+	**7.9348E-02 (9.7399E-03)**
	(1,10)	2.4005E-01 (1.3859E-02)+	2.1716E-01 (3.2408E-02)+	4.3168E-01 (1.8622E-01)+	1.8367E-01 (2.4023E-02)+	5.3119E-01 (2.0009E-02)+	**6.8815E-01 (1.9672E-02)**
DF11	(10,10)	6.5392E-01 (1.5009E-03)+	6.6180E-01 (1.9902E-02)+	6.4926E-01 (3.3183E-02)+	6.3680E-01 (1.5372E-03)+	6.8192E-01 (1.5601E-02)+	**6.0533E-01 (1.2674E-03)**
	(1,30)	5.4780E-01 (5.3343E-04)+	5.4641E-01 (1.9326E-03)≈	5.4989E-01 (8.3355E-03)+	5.4547E-01 (2.2452E-03)≈	5.9357E-01 (2.0508E-02)+	**5.4294E-01 (1.7999E-03)**
	(5,10)	6.6969E-01 (1.4498E-03)+	6.6853E-01 (5.4275E-03)+	6.6264E-01 (1.1167E-02)+	**6.5530E-01 (1.3842E-03)**≈	6.9740E-01 (1.9976E-02)+	6.5796E-01 (1.6688E-03)
	(2.5,10)	6.6405E-01 (1.5969E-03)+	6.6218E-01 (2.1266E-03)+	6.5949E-01 (3.4071E-03)+	6.5274E-01 (3.3643E-03)≈	6.8942E-01 (2.1843E-02)+	**6.5230E-01 (2.4288E-03)**
	(1,10)	5.6296E-01 (2.4306E-03)+	5.7089E-01 (2.2866E-03)+	5.6222E-01 (2.0475E-03)+	5.5278E-01 (2.8837E-03)+	6.1257E-01 (2.0387E-02)+	**5.3871E-01 (2.1252E-03)**
DF12	(10,10)	2.4673E-01 (1.0752E-02)+	2.7679E-01 (3.4366E-02)+	9.7991E-02 (5.7432E-03)+	**9.4742E-02 (4.9730E-03)**≈	1.8308E-01 (4.5885E-03)+	9.8568E-02 (8.8508E-03)
	(1,30)	2.0791E-01 (1.3720E-02)+	1.9173E-01 (2.2163E-02)+	4.5666E-01 (1.5405E-01)+	**1.5914E-01 (1.7380E-02)–**	2.0953E-01 (1.1145E-02)+	1.9431E-01 (3.6752E-02)
	(5,10)	2.9432E-01 (1.2210E-02)+	2.7948E-01 (4.1479E-02)+	1.4132E-01 (1.7557E-02)+	**1.2186E-01 (1.3165E-02)–**	2.3304E-01 (1.2045E-02)+	1.4824E-01 (5.2013E-02)
	(2.5,10)	3.1256E-01 (2.5472E-02)+	2.2647E-01 (2.5067E-02)+	2.1266E-01 (5.7831E-02)+	**1.3064E-01 (1.0389E-02)–**	2.4447E-01 (1.1946E-02)+	1.8658E-01 (4.6199E-02)
	(1,10)	2.7747E-01 (1.4096E-02)+	2.4811E-01 (1.8299E-02)+	5.7625E-01 (9.6925E-02)+	1.9865E-01 (1.8153E-02)+	3.6732E-01 (1.3155E-02)+	**1.7105E-01 (3.0346E-02)**
DF13	(10,10)	2.8903E-01 (4.8700E-03)+	3.2398E-01 (2.3680E-02)+	2.6842E-01 (2.2709E-02)+	2.6218E-01 (2.9402E-02)+	4.8524E-01 (2.0436E-02)+	**2.3759E-01 (2.2411E-02)**
	(1,30)	2.8268E-01 (6.2330E-03)+	2.6016E-01 (3.2667E-02)+	4.2312E-01 (8.9757E-02)+	2.4546E-01 (2.2903E-02)+	5.4994E-01 (2.2187E-02)+	**2.0346E-01 (2.0020E-02)**
	(5,10)	3.9076E-01 (8.8156E-02)+	3.3996E-01 (2.7335E-02)+	**2.5992E-01 (2.0633E-02)**≈	2.6567E-01 (2.3230E-02)≈	4.0584E-01 (2.1469E-02)+	2.6004E-01 (3.7639E-02)
	(2.5,10)	6.6561E-01 (1.5134E-01)+	3.4271E-01 (2.9593E-02)+	4.1306E-01 (1.2663E-01)+	2.6951E-01 (1.8662E-02)+	4.2065E-01 (1.9312E-02)+	**2.3573E-01 (1.4885E-02)**
	(1,10)	2.9602E-01 (2.8471E-02)≈	3.6518E-01 (4.1525E-02)+	1.0176E+00 (2.9139E-01)+	**2.7844E-01 (3.0885E-02)**≈	5.2320E-01 (2.5604E-02)+	2.8157E-01 (4.0587E-02)
DF14	(10,10)	7.0504E-02 (8.8086E-04)+	6.3584E-02 (2.3618E-03)+	5.9296E-02 (2.4808E-03)≈	5.8663E-02 (2.8319E-03)≈	9.0997E-02 (2.5150E-03)+	**5.7985E-02 (2.4205E-03)**
	(1,30)	5.8066E-02 (2.7807E-03)+	3.8351E-02 (1.5508E-03)≈	3.9784E-02 (2.9198E-03)+	**3.8023E-02 (1.9321E-03)**≈	6.7703E-02 (1.9785E-03)+	4.0285E-02 (3.3351E-03)
	(5,10)	7.4054E-02 (2.3397E-03)+	6.4360E-02 (1.9245E-03)+	6.4272E-02 (4.8173E-03)≈	6.1151E-02 (3.7799E-03)+	1.0347E-01 (5.1430E-03)+	**5.3400E-02 (2.3272E-03)**
	(2.5,10)	1.2704E-01 (4.2868E-02)+	6.2861E-02 (2.6633E-03)+	7.7198E-02 (1.1447E-02)+	6.1962E-02 (4.2811E-03)≈	1.4827E-01 (5.2879E-03)+	**6.0531E-02 (9.6957E-03)**
	(1,10)	7.8896E-02 (2.2538E-02)+	5.6179E-02 (6.7604E-03)≈	2.0172E-01 (7.3958E-02)+	**5.1045E-02 (7.4529E-03)**≈	1.6676E-01 (7.0337E-03)+	5.4780E-02 (8.7186E-03)
w / t / l		**57 / 8 / 5**	**56 / 14 / 0**	**63 / 7 / 0**	**37 / 29 / 4**	**66 / 3 / 1**	–

Superior performances are highlighted in bold. "+", "≈", and "–" report the statistical significance differences between medians calculated by the Wilcoxon rank sum test at a 5% significance level, which indicate that *CP-MOEA/D* is superior, approximate, and worse to the compared algorithm, respectively.
$w/t/l$ indicates that *CP-MOEA/D* wins on w instances, ties on t instances, and loses on l instances in the comparison, respectively.

to Group G_1 in Table I) are reported in Table II and Table III, respectively. Each algorithm runs independently 20 times on a total of 14 DF benchmark functions. The best results in these tables are highlighted in bold. The Wilcoxon rank-sum test is performed at a 5% significance level to analyze the differences between the compared counterparts and the proposed *CP-MOEA/D*. The symbols "+", "≈", and "–" indicate that the proposed *CP-MOEA/D* performs statistically better, shows no significant difference, or performs worse compared to the other algorithm, respectively. Additionally, the MIGD and MHV values resulting from the significance test conducted on 14 DF benchmarks across 5 dynamic configurations (comprising a total of 70 DF instances) are succinctly summarized in the bottom row of each table. The notation $w/t/l$ indicates the number of instances where *CP-MOEA/D* wins, ties, or loses, respectively, compared to the counterpart algorithm.

In terms of the fixed dynamic configurations in Group G_1, it can be observed from Table II that in the majority of the test instances, the proposed centralized prediction-based *CP-MOEA/D* obtains superior MIGD values against the compared counterparts. On DF5, DF6, DF9, DF10 and DF11, *CP-MOEA/D* surpassed its counterparts under almost all the 5 configurations of fixed dynamics. In summary, *CP-MOEA/D* obtained the best averaged MIGD values over the competitors on 44 out of a total 70 number of DF instances, while *SVR-MOEA/D*, *Tr-MOEA/D*, *AE-MOEA/D*, *MV-MOEA/D* and *MSTL-MOEA/D* obtained 8, 0, 1, 15 and 2 best MIGD results, respectively. Similar observations regarding the MHV metric can be found in the performance comparisons. In particular, *CP-MOEA/D* won 45 out of 70 cases, while the other compared counterparts obtained 6, 3, 2, 14 and 0 best MHV values, respectively. Furthermore, Wilcoxon rank sum test results (see the bottom row of "$w/t/l$" in Table II and Table III) reveal that, incorporating centralized prediction with cross-problem learning, *CP-MOEA/D* achieved significantly better MIGD values compared to its counterparts on 57, 56, 63,

TABLE III
MEAN AND STANDARD DEVIATIONS OF MHV VALUES OBTAINED BY 6 ALGORITHMS WITH 5 FIXED DYNAMIC SETTINGS

Problems	(n_t,τ_t)	*SVR-MOEA/D*	*Tr-MOEA/D*	*AE-MOEA/D*	*MV-MOEA/D*	*MSTL-MOEA/D*	*CP-MOEA/D*
DF1	(10,10)	5.2373E-01 (1.4761E-02)+	6.1907E-01 (7.3577E-03)+	5.9371E-01 (3.4589E-02)+	6.2128E-01 (6.0432E-03)+	4.8055E-01 (3.6258E-03)+	**6.4632E-01 (8.8902E-03)**
	(1,30)	6.6815E-01 (1.6123E-03)≈	6.5788E-01 (5.9228E-03)+	5.4737E-01 (5.9271E-02)+	6.5325E-01 (8.8109E-03)+	5.3594E-01 (6.8109E-03)+	**6.7751E-01 (7.6284E-03)**
	(5,10)	4.9877E-01 (1.9487E-02)+	**6.2621E-01 (2.6798E-03)**≈	5.7912E-01 (3.6724E-02)+	5.9823E-01 (5.5355E-02)+	4.7038E-01 (8.6043E-03)+	6.1495E-01 (1.8767E-02)
	(2.5,10)	4.2290E-01 (2.2503E-02)+	6.2978E-01 (1.6340E-02)≈	4.1155E-01 (6.7096E-02)+	6.2539E-01 (3.2483E-02)+	4.4480E-01 (2.0485E-02)+	**6.3671E-01 (1.3652E-02)**
	(1,10)	4.5144E-01 (1.8832E-02)+	6.2379E-01 (1.0885E-02)+	6.8376E-02 (2.5226E-02)+	6.3613E-01 (1.8164E-02)≈	3.8407E-01 (6.3255E-03)+	**6.4322E-01 (9.0238E-03)**
DF2	(10,10)	7.9607E-01 (8.4053E-03)+	**8.5146E-01 (5.5694E-03)**≈	7.7415E-01 (3.8443E-02)+	8.4843E-01 (1.0306E-02)≈	6.6772E-01 (7.0564E-02)+	8.4681E-01 (1.3015E-02)
	(1,30)	**8.6847E-01 (5.3428E-04)**–	8.5770E-01 (6.3365E-03)≈	8.2389E-01 (5.0536E-02)+	8.5743E-01 (7.0158E-03)≈	6.9864E-01 (2.3621E-03)+	8.5694E-01 (1.2491E-02)
	(5,10)	7.5473E-01 (1.2315E-02)+	8.4671E-01 (9.9321E-03)≈	7.6262E-01 (3.2912E-02)+	**8.4688E-01 (1.3598E-02)**≈	6.5605E-01 (8.6002E-03)+	8.4225E-01 (5.4060E-03)
	(2.5,10)	6.9149E-01 (1.1397E-02)+	8.3847E-01 (2.7817E-02)≈	6.9325E-01 (4.7797E-02)+	**8.3929E-01 (2.4180E-02)**≈	6.2687E-01 (2.1433E-02)+	8.3863E-01 (6.3432E-03)
	(1,10)	6.5270E-01 (3.8488E-02)+	8.2398E-01 (7.3056E-03)≈	2.2113E-01 (3.3351E-02)+	8.2583E-01 (6.5954E-03)≈	5.2499E-01 (5.2584E-03)+	**8.2643E-01 (6.1002E-03)**
DF3	(10,10)	2.1993E-01 (2.2938E-02)+	3.2316E-01 (1.5963E-02)+	4.2607E-01 (4.0342E-02)+	3.6483E-01 (9.9568E-02)+	3.7417E-01 (3.1176E-02)+	**4.3568E-01 (9.7793E-02)**
	(1,30)	3.2742E-01 (2.2583E-02)+	4.9179E-01 (1.7167E-02)+	3.5267E-01 (1.8851E-02)+	4.9915E-01 (2.6813E-02)+	4.2249E-01 (2.0458E-02)+	**5.3321E-01 (3.0383E-02)**
	(5,10)	2.2803E-01 (2.3221E-02)+	3.3623E-01 (1.1402E-02)+	3.9946E-01 (6.3541E-02)+	3.7355E-01 (7.7897E-02)+	3.3470E-01 (2.4135E-02)+	**4.3726E-01 (8.6502E-02)**
	(2.5,10)	2.7956E-01 (2.1363E-02)+	4.1904E-01 (1.7342E-02)+	3.2207E-01 (3.2428E-02)+	4.5825E-01 (4.5377E-02)+	3.2054E-01 (2.1689E-02)+	**4.7155E-01 (3.7839E-02)**
	(1,10)	2.0930E-01 (7.2492E-03)+	4.4468E-01 (9.3052E-03)+	2.2840E-01 (1.8236E-02)+	**4.9237E-01 (4.3353E-02)**≈	2.4936E-01 (1.5453E-02)+	4.8973E-01 (4.1828E-02)
DF4	(10,10)	**6.2821E+00 (1.9961E-03)**≈	6.2103E+00 (1.7822E-02)+	6.2576E+00 (3.7000E-02)≈	6.2344E+00 (3.1115E-02)≈	8.4013E-01 (2.0410E-02)+	6.2599E+00 (3.8013E-02)
	(1,30)	4.0159E+00 (9.5507E-04)+	5.4114E+00 (2.3661E-02)≈	3.9745E+00 (1.5625E-02)+	**5.4244E+00 (3.0333E-02)**≈	8.3666E-01 (3.2804E-02)+	5.4048E+00 (1.2267E-02)
	(5,10)	5.2762E+00 (5.5105E-02)≈	4.1772E+00 (2.1832E-02)+	5.3192E+00 (2.7437E-02)≈	4.2079E+00 (1.7378E-02)+	8.4675E-01 (3.1459E-02)+	**5.3312E+00 (2.9431E-02)**
	(2.5,10)	6.2061E+00 (4.1047E-02)+	6.3512E+00 (1.3391E-02)≈	6.2334E+00 (4.5006E-02)+	**6.3788E+00 (6.5162E-02)**≈	7.6676E-01 (1.0093E-02)+	6.3731E+00 (2.4890E-02)
	(1,10)	3.9099E+00 (2.6446E-02)+	4.1028E+00 (4.1517E-02)+	3.8109E+00 (5.9506E-02)+	4.1617E+00 (2.6760E-02)+	6.2230E-01 (1.8695E-02)+	**4.2374E+00 (2.1649E-02)**
DF5	(10,10)	6.8684E-01 (6.0108E-04)≈	6.8416E-01 (4.9246E-03)≈	6.8800E-01 (2.6559E-03)≈	6.8660E-01 (7.7831E-03)≈	5.4564E-01 (5.4153E-03)+	**6.9096E-01 (3.1272E-03)**
	(1,30)	7.0893E-01 (1.3828E-04)+	7.0279E-01 (3.0831E-03)≈	6.9494E-01 (1.8982E-02)≈	**7.1071E-01 (2.0003E-03)**≈	5.7232E-01 (2.8425E-03)+	7.0324E-01 (2.6310E-03)
	(5,10)	6.4756E-01 (1.5289E-02)+	6.8559E-01 (3.5444E-03)+	6.8460E-01 (2.2443E-03)+	6.9319E-01 (3.7565E-03)+	5.4183E-01 (3.0232E-03)+	**6.9737E-01 (3.6245E-03)**
	(2.5,10)	5.5861E-01 (1.6451E-02)+	6.8823E-01 (2.4182E-03)+	6.5595E-01 (6.2705E-03)+	6.9596E-01 (3.8852E-03)≈	5.1991E-01 (3.1455E-03)+	**6.9917E-01 (1.9638E-03)**
	(1,10)	6.8683E-01 (1.0770E-02)+	6.4852E-01 (1.6438E-02)+	4.8840E-01 (4.1681E-02)+	6.9779E-01 (7.4912E-03)≈	4.3475E-01 (9.7547E-03)+	**6.9804E-01 (6.4401E-03)**
DF6	(10,10)	3.4134E-01 (5.6310E-02)+	1.9296E-01 (1.9567E-02)+	4.7066E-01 (2.3010E-01)+	6.4558E-01 (5.1554E-02)+	4.8107E-01 (3.2144E-02)+	**6.7568E-01 (2.1402E-02)**
	(1,30)	1.7281E-01 (5.6372E-02)+	2.1882E-01 (1.5422E-02)+	1.2224E-01 (5.1089E-02)+	5.2539E-01 (1.6355E-02)+	1.6016E-01 (1.4507E-02)+	**6.1402E-01 (1.6466E-02)**
	(5,10)	2.2417E-01 (3.5162E-02)+	1.8436E-01 (1.8251E-02)+	4.1688E-01 (1.3751E-01)+	6.3327E-01 (2.5461E-02)≈	4.1898E-01 (2.1453E-02)+	**6.4970E-01 (1.3559E-02)**
	(2.5,10)	1.6068E-01 (3.2031E-02)+	2.0143E-01 (1.9202E-02)+	3.3515E-01 (7.8699E-02)+	6.5266E-01 (2.0974E-02)+	2.4002E-01 (2.1067E-02)+	**6.9315E-01 (1.4867E-02)**
	(1,10)	1.6045E-01 (4.1427E-02)+	1.9557E-01 (5.0532E-03)+	1.3293E-01 (4.3636E-02)+	5.2062E-01 (1.4257E-02)≈	1.7992E-01 (1.0104E-02)+	**5.3240E-01 (1.3503E-02)**
DF7	(10,10)	1.8113E+00 (4.6599E-02)+	1.6210E+00 (2.4208E-01)+	1.9015E+00 (9.9447E-02)≈	1.7703E+00 (2.8080E-01)+	2.9457E-01 (2.1436E-02)+	**2.0172E+00 (1.4082E-01)**
	(1,30)	**1.6952E+00 (3.4118E-02)**–	1.1280E+00 (1.7001E-01)–	1.1045E+00 (1.7685E-01)–	6.2424E-01 (2.4721E-01)+	3.6552E-02 (3.5330E-03)+	9.1324E-01 (1.3045E-01)
	(5,10)	**1.0686E+00 (2.5936E-01)**≈	2.6639E-01 (1.2447E-01)+	2.5973E-01 (4.0816E-01)+	4.1058E-01 (9.2626E-02)+	1.8228E-01 (4.1534E-03)+	9.9560E-01 (4.3358E-01)
	(2.5,10)	1.4477E+00 (4.6181E-02)≈	1.2983E+00 (3.4420E-02)+	1.3811E+00 (5.7658E-02)+	1.2857E+00 (2.1104E-02)+	5.3741E-02 (3.0199E-03)+	**1.5602E+00 (1.8112E-02)**
	(1,10)	**1.6341E+00 (4.0367E-02)**–	1.2472E+00 (3.6440E-02)–	9.2628E-01 (9.9897E-02)+	1.2545E+00 (1.7908E-02)–	2.5512E-02 (1.9315E-03)+	9.8040E-01 (8.2334E-02)
DF8	(10,10)	7.0482E-01 (1.7402E-03)+	7.0694E-01 (4.1334E-03)+	5.0149E-01 (1.1797E-01)+	6.8926E-01 (2.2444E-02)+	5.9933E-01 (4.7411E-03)+	**7.2656E-01 (1.9331E-02)**
	(1,30)	8.8819E-01 (7.3001E-04)+	**8.9781E-01 (1.9435E-03)**≈	8.4383E-01 (2.9025E-02)+	8.9059E-01 (1.0430E-02)≈	7.4172E-01 (1.9506E-03)+	8.9699E-01 (2.3325E-02)
	(5,10)	**7.0570E-01 (1.5284E-03)**–	7.0546E-01 (4.6114E-03)–	5.8269E-01 (7.7716E-02)+	6.7304E-01 (1.8010E-02)+	6.0063E-01 (2.0800E-03)+	6.8280E-01 (9.1716E-03)
	(2.5,10)	7.0659E-01 (1.4110E-03)+	7.0603E-01 (3.0783E-03)+	5.6548E-01 (5.8802E-02)+	6.7625E-01 (1.2457E-02)+	5.9772E-01 (2.1451E-03)+	**7.2442E-01 (1.7996E-02)**
	(1,10)	8.7706E-01 (2.4068E-03)+	8.9207E-01 (3.6334E-03)≈	8.4823E-01 (2.9204E-02)+	8.7060E-01 (3.0571E-02)+	7.3592E-01 (3.2657E-03)+	**8.9220E-01 (2.3050E-02)**
DF9	(10,10)	3.9862E-01 (5.8457E-02)+	5.7800E-01 (1.6759E-02)+	3.5153E-01 (4.8438E-02)+	6.3674E-01 (8.2942E-03)+	4.3702E-01 (5.6048E-03)+	**7.3313E-01 (4.7438E-03)**
	(1,30)	4.7875E-01 (5.2959E-02)+	5.8282E-01 (1.2676E-02)+	3.0184E-01 (2.8409E-02)+	6.1937E-01 (3.2419E-03)+	3.7665E-01 (3.4639E-03)+	**7.1182E-01 (3.8344E-03)**
	(5,10)	3.7188E-01 (5.9507E-02)+	5.7150E-01 (1.3002E-02)+	2.5505E-01 (3.6104E-02)+	6.2116E-01 (7.5522E-03)≈	3.8657E-01 (3.4795E-03)+	**6.2667E-01 (9.2578E-03)**
	(2.5,10)	4.1193E-01 (4.0494E-02)+	5.6952E-01 (1.4446E-02)+	1.4134E-01 (2.4427E-02)+	6.1329E-01 (5.5256E-03)+	3.2329E-01 (5.0302E-03)+	**6.1824E-01 (7.8240E-03)**
	(1,10)	3.1907E-01 (4.3196E-02)+	5.4375E-01 (9.6303E-03)+	1.1545E-01 (2.0442E-02)+	5.8984E-01 (5.9281E-03)≈	2.7319E-01 (5.1488E-03)+	**5.9342E-01 (2.5785E-03)**
DF10	(10,10)	8.1039E-01 (1.3974E-02)≈	7.9660E-01 (8.1395E-03)+	6.9105E-01 (5.0126E-02)+	8.1349E-01 (9.2091E-03)≈	4.7115E-01 (4.5196E-03)+	**8.1720E-01 (7.0198E-03)**
	(1,30)	7.9659E-01 (1.3974E-02)+	9.2230E-01 (1.6409E-02)+	9.1810E-01 (6.9003E-02)+	9.5779E-01 (1.1433E-02)+	6.1569E-01 (6.3554E-03)+	**9.6682E-01 (1.3959E-02)**
	(5,10)	8.5372E-01 (1.3974E-02)+	9.1303E-01 (6.6803E-03)+	8.0740E-01 (1.7801E-01)+	9.2328E-01 (9.9409E-03)+	5.8987E-01 (6.2547E-03)+	**9.3462E-01 (8.8297E-03)**
	(2.5,10)	7.3357E-01 (1.3974E-02)+	8.7508E-01 (1.0446E-02)+	8.4201E-01 (1.1896E-01)+	8.9714E-01 (1.5210E-02)≈	5.4951E-01 (1.0483E-02)+	**8.9965E-01 (1.1376E-02)**
	(1,10)	9.0516E-01 (1.3974E-02)≈	8.8795E-01 (1.8287E-02)≈	6.2815E-01 (1.5783E-01)+	**9.1804E-01 (1.3935E-02)**–	5.7198E-01 (8.7999E-03)+	8.6863E-01 (1.2182E-02)
DF11	(10,10)	2.0019E-02 (6.8076E-03)+	7.8127E-02 (1.5401E-03)+	**9.2878E-02 (2.7214E-03)**≈	7.8982E-02 (1.7457E-03)+	6.3372E-02 (1.4581E-03)+	8.9576E-02 (6.2009E-03)
	(1,30)	1.0469E-01 (2.1606E-03)+	1.7508E-01 (4.7311E-03)+	3.4158E-01 (6.5359E-03)≈	1.7579E-01 (3.7095E-03)+	2.4532E-01 (3.0366E-03)+	**3.4942E-01 (5.3785E-03)**
	(5,10)	1.5892E-02 (2.4922E-03)+	6.7280E-02 (1.7527E-03)+	7.7918E-02 (2.4226E-03)≈	7.0930E-02 (1.7749E-03)+	4.8939E-02 (2.4656E-03)+	**7.8297E-02 (2.1050E-03)**
	(2.5,10)	1.3375E-02 (2.5649E-03)+	7.3594E-02 (3.3759E-03)+	7.7778E-02 (3.1360E-03)+	**8.0941E-02 (2.0189E-03)**≈	6.0242E-02 (2.3195E-03)+	8.0621E-02 (1.8190E-03)
	(1,10)	9.4255E-02 (3.8707E-03)+	2.1992E-01 (5.4286E-03)+	**3.3956E-01 (4.5301E-03)**≈	2.4526E-01 (3.1863E-03)+	1.7683E-01 (3.9559E-03)+	3.3732E-01 (1.1728E-02)
DF12	(10,10)	8.2813E-01 (3.6454E-03)+	1.6365E+00 (5.2371E-02)+	1.7515E+00 (1.0520E-02)≈	**1.8639E+00 (1.2190E-02)**–	5.6751E-01 (2.1884E-02)+	1.7463E+00 (9.1052E-03)
	(1,30)	9.5476E-01 (1.5966E-02)+	1.7419E+00 (4.6139E-02)≈	1.3238E+00 (1.7777E-01)+	**1.7774E+00 (3.2023E-02)**≈	3.7132E-01 (1.9460E-02)+	1.6970E+00 (3.5184E-02)
	(5,10)	9.2161E-01 (4.2833E-03)+	1.5763E+00 (5.9104E-02)+	1.7705E+00 (5.9375E-02)+	1.8227E+00 (2.6168E-02)+	4.5638E-01 (2.0251E-02)+	**1.8960E+00 (1.3105E-02)**
	(2.5,10)	7.9369E-01 (7.3184E-03)+	1.5477E+00 (1.6459E-02)+	1.6858E+00 (1.0527E-01)+	1.7389E+00 (1.7863E-02)+	4.7707E-01 (1.8857E-02)+	**1.8856E+00 (8.7664E-03)**
	(1,10)	9.4809E-01 (1.6029E-02)+	1.4561E+00 (3.0342E-02)+	9.5614E-01 (5.8070E-02)+	1.5672E+00 (3.4331E-02)≈	4.1109E-01 (2.5116E-02)+	**1.6366E+00 (2.5393E-02)**
DF13	(10,10)	4.9550E-01 (1.8005E-02)+	2.8263E+00 (1.4909E-01)+	3.2746E+00 (8.7652E-02)≈	3.2854E+00 (1.4879E-01)≈	5.5442E-01 (7.3848E-02)+	**3.2935E+00 (1.8321E-01)**
	(1,30)	5.5541E-01 (2.5623E-02)+	3.1436E+00 (8.4676E-02)≈	2.6354E+00 (3.1675E-01)+	**3.1626E+00 (4.2599E-02)**≈	5.6211E-01 (5.6004E-02)+	3.1590E+00 (9.6766E-02)
	(5,10)	4.7131E-01 (1.3400E-02)+	2.8956E+00 (1.2037E-01)+	3.3236E+00 (5.4901E-02)≈	**3.3859E+00 (5.8240E-02)**–	5.6512E-01 (5.1586E-02)+	3.2934E+00 (1.3342E-01)
	(2.5,10)	4.2223E-01 (1.9916E-02)+	2.8438E+00 (1.4832E-01)+	2.9038E+00 (4.0374E-01)+	3.3105E+00 (7.3955E-02)≈	5.4028E-01 (5.2340E-02)+	**3.3637E+00 (1.1807E-01)**
	(1,10)	5.4497E-01 (4.1057E-02)+	2.5599E+00 (1.6503E-01)+	1.2236E+00 (2.1794E-01)+	2.9176E+00 (1.5575E-01)≈	4.4140E-01 (5.1004E-01)+	**2.9504E+00 (1.1959E-01)**
DF14	(10,10)	5.7375E-02 (2.1490E-03)+	5.0727E-01 (6.6709E-03)+	5.2009E-01 (1.2601E-02)+	**5.2899E-01 (5.7737E-03)**≈	2.8483E-01 (5.1657E-03)+	5.2837E-01 (6.2556E-03)
	(1,30)	6.8745E-04 (1.3212E-07)+	4.5410E-01 (3.5154E-03)+	5.4600E-01 (6.8128E-03)≈	4.5292E-01 (2.9816E-03)+	1.3262E-01 (3.6468E-03)+	**5.4842E-01 (6.1099E-03)**
	(5,10)	1.0657E-01 (1.0855E-03)+	4.9404E-01 (5.5462E-03)+	5.0278E-01 (1.5979E-02)+	5.1835E-01 (3.9965E-03)≈	3.6580E-01 (5.0497E-03)+	**5.1971E-01 (5.7549E-03)**
	(2.5,10)	7.9574E-02 (1.5094E-03)+	4.7310E-01 (5.4635E-03)≈	4.7179E-01 (1.1413E-02)+	**4.8547E-01 (2.9610E-03)**≈	4.2249E-01 (5.1349E-03)+	4.8034E-01 (6.9441E-03)
	(1,10)	2.4751E-03 (3.7732E-03)+	4.6888E-01 (8.1961E-03)+	3.8180E-01 (6.7692E-02)+	4.7701E-01 (5.8228E-03)+	2.2688E-01 (3.8566E-03)+	**5.2797E-01 (1.2241E-02)**
w / t / l		58 / 8 / 4	50 / 17 / 3	56 / 13 / 1	35 / 31 / 4	70 / 0 / 0	–

Superior performances are highlighted in bold. "+", "≈", and "–" report the statistical significance differences between medians calculated by the Wilcoxon rank sum test at a 5% significance level, which indicate that *CP-MOEA/D* is superior, approximate, and worse to the compared algorithm, respectively.
$w/t/l$ indicates that *CP-MOEA/D* wins on w instances, ties on t instances, and loses on l instances in the comparison, respectively.

37 and 66 instances, respectively. Additionally, *CP-MOEA/D* outperformed its counterparts significantly on the MHV metric on 58, 50, 56, 35 and 70 instances, respectively. Since the number of the tested problem instances is large, the Friedman test ranks in terms of MIGD and MHV values are further calculated and presented in Fig. 3 to summarize the overall performances. It can be observed that the ranking scores in terms of the MIGD and MHV metrics for *CP-MOEA/D* (1.90 and 1.64) are significantly better than *SVR-MOEA/D* (4.26 and 4.39), *Tr-MOEA/D* (3.41 and 3.36), *AE-MOEA/D* (4.57 and 4.01), and the recently proposed *MSTL-MOEA/D* (4.35 and 5.29). *CP-MOEA/D* also outperforms the state-of-the-art *MV-MOEA/D* (2.52 and 2.31), which relies on multi-view prediction within a single problem domain. Therefore, the algorithm ranking results are mainly consistent with the tabular results obtained on DMIGD and DMHV metrics, respectively.

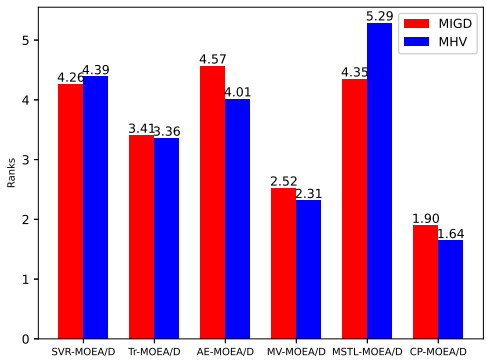


Fig. 3. Friedman test ranks of MIGD and MHV instances for all the compared algorithms on 70 DF instances in Group G_1.

In terms of the modified shifty dynamic configurations in

TABLE IV
MEAN AND STANDARD DEVIATIONS OF MIGD VALUES OBTAINED BY 6 ALGORITHMS WITH 2 SHIFTY DYNAMIC SETTINGS

Problems	(n_t,τ_t)	*SVR-MOEA/D*	*Tr-MOEA/D*	*AE-MOEA/D*	*MV-MOEA/D*	*MSTL-MOEA/D*	*CP-MOEA/D*
DF1	([1,5,10],10)	1.7562E-01 (4.5601E-02)+	1.0328E-01 (6.5284E-02)+	2.3447E-01 (6.7641E-02)+	1.4692E-01 (2.6849E-02)+	1.2346E-01 (2.0157E-02)+	**4.3570E-02 (5.1248E-03)**
	([5,[10,30])	**3.6902E-03 (2.0691E-03)**≈	3.5382E-02 (9.4681E-03)+	7.1490E-02 (2.5763E-02)+	9.3846E-02 (6.5281E-03)+	2.8468E-02 (3.5486E-03)+	7.5579E-03 (1.2104E-03)
DF2	([1,5,10],10)	2.8306E-01 (5.2755E-02)+	8.3008E-02 (4.7241E-02)+	1.6570E-01 (4.2519E-02)+	5.4333E-02 (1.6091E-02)+	7.9939E-02 (1.8046E-02)+	**2.4586E-02 (7.8915E-03)**
	([5,[10,30])	5.4506E-03 (8.4187E-04)≈	5.6802E-02 (9.5114E-03)+	4.0159E-02 (2.0257E-02)+	1.0091E-02 (6.2084E-03)+	4.8329E-02 (5.2469E-03)+	**4.8894E-03 (1.3515E-03)**
DF3	([1,5,10],10)	9.3019E-01 (8.5579E-02)+	7.5861E-01 (5.4751E-02)+	4.0955E-01 (9.7426E-02)+	3.9043E-01 (1.4133E-01)+	6.8479E-01 (8.4950E-02)+	**1.5698E-01 (5.0498E-02)**
	([5,[10,30])	3.0236E-01 (2.7574E-02)+	1.6048E-01 (1.9920E-02)+	2.7714E-01 (2.0885E-02)+	9.5344E-02 (4.5831E-02)≈	1.3517E-01 (4.5831E-02)+	**8.9431E-02 (1.0886E-02)**
DF4	([1,5,10],10)	4.3897E-01 (5.5740E-03)+	3.5426E-01 (3.6431E-02)+	6.5660E-01 (8.3541E-02)+	3.2809E-01 (6.7743E-02)+	2.8479E-01 (4.5600E-02)+	**7.0092E-02 (2.5539E-02)**
	([5,[10,30])	7.6531E-02 (6.1549E-03)+	8.4091E-02 (2.6493E-02)+	1.2538E-01 (3.5164E-02)+	7.0106E-02 (8.2440E-03)+	7.8515E-02 (7.8387E-03)+	**5.3379E-02 (2.0207E-02)**
DF5	([1,5,10],10)	2.9164E-01 (2.5008E-03)+	1.0177E-01 (2.7835E-02)+	8.5942E-02 (7.6145E-03)+	7.5826E-02 (6.5163E-03)+	9.4693E-02 (7.2547E-03)+	**1.9554E-02 (3.4870E-03)**
	([5,[10,30])	2.3051E-02 (4.0473E-03)+	6.7430E-02 (4.5218E-03)+	7.5609E-02 (2.3598E-02)+	2.6997E-02 (2.6042E-03)+	5.6566E-02 (2.8348E-03)+	**1.6473E-02 (6.4251E-03)**
DF6	([1,5,10],10)	5.2061E+00 (5.4957E-01)+	1.6995E+00 (4.2583E-01)+	1.2468E+00 (1.8680E+00)+	9.5249E-01 (2.4552E-01)+	9.0146E-01 (2.0322E-01)+	**3.2641E-01 (8.2011E-02)**
	([5,[10,30])	5.7949E-01 (5.2743E-02)+	4.5433E-01 (3.2840E-02)+	5.0581E-01 (7.2473E-02)+	4.7580E-01 (3.8265E-02)+	4.6086E-01 (5.2489E-02)+	**4.1984E-01 (5.0448E-02)**
DF7	([1,5,10],10)	1.8319E+00 (4.5460E-02)+	1.3507E+00 (6.9605E-02)+	8.0493E-01 (9.2417E-02)+	7.5127E-01 (5.3629E-02)+	1.3056E+00 (5.9491E-02)+	**5.7493E-01 (3.1316E-02)**
	([5,[10,30])	8.0670E-01 (6.9812E-02)+	1.3015E+00 (2.2584E-01)+	1.5896E+00 (2.175E-01)+	1.4260E+00 (2.3005E-01)+	1.2168E+00 (6.8461E-02)+	**7.1884E-01 (3.1684E-02)**
DF8	([1,5,10],10)	1.8519E-01 (8.3046E-03)+	9.9158E-02 (1.5420E-02)≈	4.6049E-01 (7.4560E-02)+	2.8774E-01 (3.9614E-02)+	1.1455E-01 (3.1456E-02)+	**7.5802E-02 (2.0435E-02)**
	([5,[10,30])	8.6249E-02 (2.0849E-03)+	1.4321E-01 (4.5346E-02)+	2.0358E-01 (5.2841E-02)+	1.2884E-01 (3.0643E-02)+	1.1564E-01 (3.1872E-02)+	**6.8496E-02 (4.2187E-03)**
DF9	([1,5,10],10)	8.4135E-01 (7.3517E-02)+	5.4259E-01 (6.0453E-02)+	5.8407E-01 (9.7643E-02)+	2.2465E-01 (5.2110E-02)+	5.4385E-01 (4.5419E-02)+	**1.1459E-01 (2.1983E-02)**
	([5,[10,30])	2.4391E-01 (3.2418E-02)+	2.5115E-01 (2.2554E-02)+	3.9652E-01 (5.2477E-02)+	**1.3954E-01 (3.6579E-03)**≈	2.3750E-01 (3.4761E-03)+	1.5643E-01 (5.4865E-03)
DF10	([1,5,10],10)	3.5880E-01 (1.6142E-02)+	2.5426E-01 (2.1345E-02)+	3.8651E-01 (3.4513E-02)+	2.0455E-01 (1.9966E-02)+	2.1881E-01 (1.8473E-02)+	**1.6154E-01 (7.8541E-03)**
	([5,[10,30])	2.7248E-01 (1.5116E-02)+	3.5864E-01 (5.2548E-02)+	2.5919E-01 (4.2596E-02)+	1.9633E-01 (1.7918E-02)+	2.7652E-01 (1.9549E-02)+	**1.4260E-01 (5.6680E-03)**
DF11	([1,5,10],10)	7.3126E-01 (1.9342E-03)+	6.7932E-01 (1.9497E-02)+	6.8861E-01 (3.2109E-02)+	6.5147E-01 (1.9980E-03)+	6.5147E-01 (1.9980E-03)+	**5.1275E-01 (4.5298E-03)**
	([5,[10,30])	5.5231E-01 (7.4239E-04)+	5.5054E-01 (2.1544E-03)≈	**5.4561E-01 (8.0796E-03)**≈	5.4594E-01 (2.3360E-03)≈	5.4822E-01 (2.4533E-03)≈	5.4621E-01 (3.6342E-03)
DF12	([1,5,10],10)	4.5493E-01 (2.4266E-02)+	5.6593E-01 (3.5942E-02)+	4.1523E-01 (2.1554E-02)+	3.3865E-01 (3.4928E-02)+	5.2697E-01 (3.0460E-02)+	**3.0116E-01 (4.2508E-02)**
	([5,[10,30])	3.5842E-01 (2.0663E-02)+	3.9259E-01 (2.4677E-02)+	3.6360E-01 (6.4825E-02)+	2.6064E-01 (1.8268E-02)+	4.3164E-01 (1.9993E-02)+	**2.5743E-01 (4.0431E-02)**
DF13	([1,5,10],10)	3.7194E-01 (8.5156E-03)+	5.1905E-01 (3.4182E-02)+	7.3218E-01 (6.2166E-02)+	5.8543E-01 (5.2299E-02)+	5.2458E-01 (5.0469E-02)+	**2.5458E-01 (9.1354E-03)**
	([5,[10,30])	2.9543E-01 (6.6300E-03)+	2.8324E-01 (2.7351E-02)+	2.5245E-01 (5.7843E-02)≈	**2.1630E-01 (2.8473E-02)–**	2.6098E-01 (2.5497E-02)+	2.4967E-01 (6.2497E-03)
DF14	([1,5,10],10)	9.1517E-02 (3.6248E-03)+	3.5817E-01 (4.5150E-02)+	1.0036E-01 (7.2805E-03)+	6.2771E-02 (3.0413E-03)≈	2.4597E-01 (3.5469E-02)+	**5.8776E-02 (3.5885E-03)**
	([5,[10,30])	5.2775E-02 (2.6608E-03)+	6.3219E-02 (4.1554E-03)+	4.5194E-02 (2.9198E-03)≈	4.6337E-02 (1.9079E-03)≈	5.5873E-02 (1.8543E-03)+	**4.4906E-02 (4.0696E-03)**
w / t / l		**26 / 2 / 0**	**26 / 2 / 0**	**25 / 3 / 0**	**22 / 5 / 1**	**27 / 1 / 0**	–

Superior performances are highlighted in bold. "+", "≈", and "–" report the statistical significance differences between medians calculated by the Wilcoxon rank sum test at a 5% significance level, which indicate that *CP-MOEA/D* is superior, approximate, and worse to the compared algorithm, respectively.
$w/t/l$ indicates that *CP-MOEA/D* wins on w instances, ties on t instances, and loses on l instances in the comparison, respectively.

TABLE V
MEAN AND STANDARD DEVIATIONS OF MHV VALUES OBTAINED BY 6 ALGORITHMS WITH 2 SHIFTY DYNAMIC SETTINGS

Problems	(n_t,τ_t)	*SVR-MOEA/D*	*Tr-MOEA/D*	*AE-MOEA/D*	*MV-MOEA/D*	*MSTL-MOEA/D*	*CP-MOEA/D*
DF1	([1,5,10],10)	3.6795E-01 (2.7149E-02)+	3.4268E-01 (4.0667E-02)+	3.9390E-01 (6.5201E-02)+	4.2643E-01 (4.5288E-02)+	3.8251E-01 (3.6374E-02)+	**5.8500E-01 (1.5178E-02)**
	([5,[10,30])	4.8816E-01 (2.0418E-02)+	4.2857E-01 (1.8750E-02)+	5.0661E-01 (2.3068E-02)≈	4.9652E-01 (6.3255E-02)≈	4.9239E-01 (3.0981E-02)≈	**5.1641E-01 (3.2776E-02)**
DF2	([1,5,10],10)	5.1854E-01 (1.3372E-02)+	5.4293E-01 (7.6021E-02)+	4.8324E-01 (2.9911E-02)+	6.0315E-01 (1.2541E-02)+	5.8076E-01 (4.0482E-02)+	**6.6734E-01 (1.4214E-02)**
	([5,[10,30])	6.6830E-01 (3.5126E-02)+	8.0057E-01 (8.4164E-02)≈	5.2536E-01 (3.7517E-02)+	**8.4660E-01 (5.3965E-02)**≈	7.4539E-01 (4.0564E-02)+	8.3059E-01 (4.0517E-02)
DF3	([1,5,10],10)	1.9255E-01 (1.6531E-02)+	3.0515E-01 (1.5146E-02)+	3.2111E-01 (3.8068E-02)+	3.7425E-01 (5.2017E-02)+	2.6096E-01 (2.8843E-02)+	**4.0310E-01 (2.0796E-02)**
	([5,[10,30])	2.9622E-01 (8.7654E-03)+	3.5043E-01 (9.2487E-03)+	2.9300E-01 (2.6463E-02)+	3.6492E-01 (4.8877E-02)+	3.1452E-01 (3.5814E-02)+	**4.5461E-01 (2.9142E-02)**
DF4	([1,5,10],10)	3.3518E+00 (5.7718E-03)+	1.0714E+00 (1.0204E-02)+	2.2267E+00 (2.4688E-02)+	3.1039E+00 (2.8860E-02)+	3.2279E+00 (1.8418E-02)+	**3.5296E+00 (3.2677E-02)**
	([5,[10,30])	4.7344E+00 (1.5210E-02)+	2.8212E+00 (3.1834E-02)+	3.6643E+00 (5.0318E-02)+	4.4142E+00 (2.0806E-02)+	4.0651E+00 (2.7241E-02)+	**5.6420E+00 (3.5554E-02)**
DF5	([1,5,10],10)	5.4219E-01 (1.7506E-03)+	4.3886E-01 (4.0825E-03)+	3.1221E-01 (1.4750E-03)+	4.6143E-01 (8.2510E-03)+	4.9521E-01 (2.0384E-02)+	**6.2886E-01 (1.1940E-02)**
	([5,[10,30])	6.6955E-01 (8.5445E-03)≈	6.0313E-01 (1.9943E-02)+	5.1227E-01 (3.3546E-02)+	6.5150E-01 (7.2645E-03)≈	6.1743E-01 (1.8089E-02)+	**6.9006E-01 (3.2803E-02)**
DF6	([1,5,10],10)	2.7126E-01 (3.2164E-02)+	3.1408E-01 (2.3335E-02)+	4.2550E-01 (4.2010E-02)+	3.6914E-01 (4.7506E-02)+	3.0255E-01 (2.7428E-02)+	**5.0464E-01 (3.0960E-02)**
	([5,[10,30])	2.2264E-01 (2.7534E-02)+	3.9641E-01 (9.8507E-03)+	1.7658E-01 (2.2167E-02)+	4.7705E-01 (3.4561E-02)+	3.9948E-01 (1.5049E-02)+	**5.6464E-01 (3.1920E-02)**
DF7	([1,5,10],10)	7.2463E-01 (3.9360E-02)+	8.2548E-01 (9.5548E-02)+	3.5249E-01 (5.7733E-02)+	1.0510E+00 (1.3062E-01)+	1.2116E+00 (1.8050E-01)+	**1.7444E+00 (3.3093E-01)**
	([5,[10,30])	1.0088E+00 (4.8325E-02)+	9.4774E-01 (7.2601E-02)+	7.8461E-01 (8.3379E-02)+	**1.3515E+00 (3.1511E-02)–**	1.1226E+00 (5.2306E-02)≈	1.2116E+00 (7.3833E-02)
DF8	([1,5,10],10)	5.5482E-01 (3.5812E-03)+	5.1337E-01 (3.7064E-03)+	4.6848E-01 (4.2860E-02)+	5.7072E-01 (2.7339E-02)+	5.2881E-01 (1.0052E-02)+	**6.0315E-01 (1.7063E-02)**
	([5,[10,30])	6.5281E-01 (3.5400E-03)+	6.8386E-01 (3.7965E-03)≈	6.0308E-01 (1.9355E-02)+	6.9051E-01 (2.8214E-02)≈	6.6063E-01 (1.8465E-02)+	**7.0037E-01 (3.6095E-02)**
DF9	([1,5,10],10)	3.2466E-01 (4.0016E-02)+	4.6091E-01 (2.3846E-02)+	4.0660E-01 (4.5278E-02)+	5.2361E-01 (7.4838E-03)+	4.3284E-01 (1.8800E-02)+	**5.7625E-01 (9.4294E-02)**
	([5,[10,30])	5.3554E-01 (5.6458E-02)+	5.6864E-01 (1.2447E-02)+	5.1081E-01 (4.3379E-02)+	6.0355E-01 (1.2160E-02)≈	5.3881E-01 (2.0426E-02)+	**6.0985E-01 (2.9110E-02)**
DF10	([1,5,10],10)	6.2544E-01 (3.5552E-02)+	5.4251E-01 (9.0568E-03)+	5.0240E-01 (3.8711E-02)+	5.2864E-01 (1.0304E-03)+	5.5744E-01 (8.8539E-03)+	**6.6577E-01 (3.6781E-02)**
	([5,[10,30])	7.5882E-01 (3.4973E-02)+	7.5041E-01 (1.2155E-02)+	6.3319E-01 (3.0212E-02)+	7.2441E-01 (2.3356E-03)+	7.0651E-01 (3.0376E-02)+	**8.2114E-01 (5.5607E-02)**
DF11	([1,5,10],10)	2.6318E-02 (5.3571E-03)+	5.5350E-02 (3.2889E-03)+	9.7903E-02 (4.5267E-03)≈	6.3031E-02 (1.6743E-03)+	6.0308E-02 (3.2886E-03)+	**9.9305E-02 (3.5473E-03)**
	([5,[10,30])	1.3654E-01 (3.2330E-03)+	2.5455E-01 (5.8561E-03)+	**3.1498E-01 (4.9285E-03)**≈	2.6314E-01 (3.2472E-03)+	2.9580E-01 (5.1477E-03)≈	2.9009E-01 (6.7860E-03)
DF12	([1,5,10],10)	7.1548E-01 (2.5682E-03)+	7.0209E-01 (3.7177E-02)+	7.5617E-01 (1.0520E-02)+	7.4599E-01 (1.5220E-02)+	7.2503E-01 (1.8673E-02)+	**8.3317E-01 (2.9097E-02)**
	([5,[10,30])	8.2165E-01 (2.4125E-02)+	8.3128E-01 (3.5049E-02)+	9.0882E-01 (3.4562E-02)+	8.8361E-01 (1.8546E-02)+	8.6425E-01 (2.8316E-02)+	**1.0218E+00 (9.0631E-02)**
DF13	([1,5,10],10)	4.4152E-01 (2.5518E-02)+	4.0503E-01 (3.8647E-02)+	5.2190E-01 (6.5713E-02)+	5.6627E-01 (4.8400E-02)+	4.3545E-01 (3.5280E-02)+	**5.9964E-01 (6.9203E-02)**
	([5,[10,30])	5.7641E-01 (2.1044E-02)+	7.3495E-01 (5.2002E-02)+	6.0248E-01 (5.4869E-02)+	7.5135E-01 (7.2381E-02)+	7.6246E-01 (6.3082E-02)+	**9.1006E-01 (8.2429E-02)**
DF14	([1,5,10],10)	2.2127E-01 (2.0894E-03)+	4.3309E-01 (6.2169E-03)+	4.4615E-01 (1.4972E-02)+	5.1517E-01 (6.3410E-03)+	4.9332E-01 (8.0746E-03)+	**6.0435E-01 (4.5557E-02)**
	([5,[10,30])	2.5990E-01 (2.7056E-03)+	5.3595E-01 (8.2762E-03)+	4.2640E-01 (1.6718E-02)+	6.4834E-01 (6.1475E-03)≈	5.2493E-01 (4.3696E-03)+	**6.6755E-01 (4.2757E-02)**
w / t / l		**27 / 1 / 0**	**26 / 2 / 0**	**25 / 3 / 0**	**21 / 6 / 1**	**25 / 3 / 0**	–

Superior performances are highlighted in bold. "+", "≈", and "–" report the statistical significance differences between medians calculated by the Wilcoxon rank sum test at a 5% significance level, which indicate that *CP-MOEA/D* is superior, approximate, and worse to the compared algorithm, respectively.
$w/t/l$ indicates that *CP-MOEA/D* wins on w instances, ties on t instances, and loses on l instances in the comparison, respectively.

Group G_2, it can be observed from Table IV that, the proposed *CP-MOEA/D* demonstrates more significant superiority compared to five state-of-the-art competitors when encountering irregular dynamics that not always predictable in the given DMOP. Specifically, for the used evaluation metrics, *CP-MOEA/D* obtained the best MIGD results on 24 out of 28 problem instances, as well as the best MHV results on 25 out of 28 problem instances, respectively. The significance test shows that *CP-MOEA/D* got significantly better MIGD values compared to its counterparts on 26, 26, 25, 22 and 27 instances; *CP-MOEA/D* also significantly surpassed them on 27, 26, 25, 21 and 25 instances with respect to the MHV metric. Note that our modified shifty dynamic settings are designed to simulate complex real-world dynamics more closely. The experimental results collected from DMOP instances of Group G_2 not only showcased the excellent performance of *CP-MOEA/D* in efficiently addressing dynamics with different degrees of change severity and change frequencies, but also exhibited a significant advantage in solving DMOPs with uncertain change patterns compared to existing learning and

TABLE VI
DMIGD AND DMHV VALUES OF THE COMPARED ALGORITHMS (SUPERIOR PERFORMANCES ARE HIGHLIGHTED IN BOLD)

	DMIGD						DMHV					
	SVR-MOEA/D	*Tr-MOEA/D*	*AE-MOEA/D*	*MV-MOEA/D*	*MSTL-MOEA/D*	*CP-MOEA/D*	*SVR-MOEA/D*	*Tr-MOEA/D*	*AE-MOEA/D*	*MV-MOEA/D*	*MSTL-MOEA/D*	*CP-MOEA/D*
DF1	1.3182E-01	4.2453E-02	2.4885E-01	5.5914E-02	9.5278E-02	**2.7998E-02**	4.8873E-01	5.6114E-01	4.4295E-01	5.7960E-01	4.5581E-01	**6.1716E-01**
DF2	1.1104E-01	4.5328E-02	1.7710E-01	3.0933E-02	6.0126E-02	**2.3003E-02**	7.0719E-01	7.9455E-01	6.1195E-01	8.0966E-01	6.4292E-01	**8.1557E-01**
DF3	4.7642E-01	2.7704E-01	3.1734E-01	2.0024E-01	2.5723E-01	**1.4300E-01**	2.5043E-01	3.8150E-01	3.3468E-01	4.1819E-01	3.2525E-01	**4.6073E-01**
DF4	1.3750E-01	1.2756E-01	1.8097E-01	1.1980E-01	4.3813E-01	**6.3659E-02**	4.8252E+00	4.3065E+00	4.4981E+00	4.8465E+00	1.6008E+00	**5.2540E+00**
DF5	8.5079E-02	3.7512E-02	7.7695E-02	2.3259E-02	6.2426E-02	**1.2303E-02**	6.4293E-01	6.3590E-01	5.7662E-01	6.5674E-01	5.3244E-01	**6.8681E-01**
DF6	2.2902E+00	8.4420E-01	1.5153E+00	5.0433E-01	6.1324E-01	**3.1395E-01**	2.2191E-01	2.4338E-01	2.9713E-01	5.4624E-01	3.1174E-01	**6.0489E-01**
DF7	1.0275E+00	1.0576E+00	1.3579E+00	1.3186E+00	2.0632E+00	**8.1952E-01**	1.3415E+00	1.0477E+00	9.5860E-01	1.1068E+00	4.1812E-01	**1.3461E+00**
DF8	9.2000E-02	1.0809E-01	2.9593E-01	1.9616E-01	1.4212E-01	**7.8770E-02**	7.2714E-01	7.2936E-01	6.3047E-01	7.2300E-01	6.3782E-01	**7.4664E-01**
DF9	3.3509E-01	1.5934E-01	6.9033E-01	**7.5690E-02**	2.8666E-01	7.6244E-02	4.0578E-01	5.5359E-01	2.9752E-01	6.0108E-01	3.9548E-01	**6.3848E-01**
DF10	2.4885E-01	2.1516E-01	2.8380E-01	**1.6525E-01**	3.5523E-01	2.0946E-01	7.8338E-01	8.1255E-01	7.1747E-01	8.2326E-01	5.8031E-01	**8.5340E-01**
DF11	6.2600E-01	6.1995E-01	6.1682E-01	6.0579E-01	6.3922E-01	**5.7946E-01**	5.8727E-02	1.3199E-01	**1.9180E-01**	1.3972E-01	1.3583E-01	1.8923E-01
DF12	3.0748E-01	3.1159E-01	3.2182E-01	**1.8633E-01**	3.1372E-01	1.9391E-01	8.5477E-01	1.3560E+00	1.3075E+00	1.4857E+00	5.5324E-01	**1.5309E+00**
DF13	3.7021E-01	3.4775E-01	4.8096E-01	3.0328E-01	4.5292E-01	**2.4609E-01**	5.0105E-01	2.2013E+00	2.0693E+00	2.4828E+00	5.5161E-01	**2.5100E+00**
DF14	7.8979E-02	1.0096E-01	8.3975E-02	5.4279E-02	1.2558E-01	**5.2952E-02**	1.0398E-01	4.8092E-01	4.7072E-01	5.1804E-01	3.5012E-01	**5.5382E-01**

prediction-based methods.

Last but not least, to investigate the robustness of the proposed centralized prediction in handling different dynamic scenarios, Table VI presents the DMIGD and DMHV values of the compared algorithms on 14 DF benchmarks containing all the 7 dynamic configurations (i.e., a total of 98 test instances), respectively. It can be observed that the proposed *CP-MOEA/D* obtained the best DMIGD and DMHV values on 11 and 13 benchmark problems, which have confirmed the robust prediction performances of the centralized prediction in solving DMOPs with diverse dynamic properties.

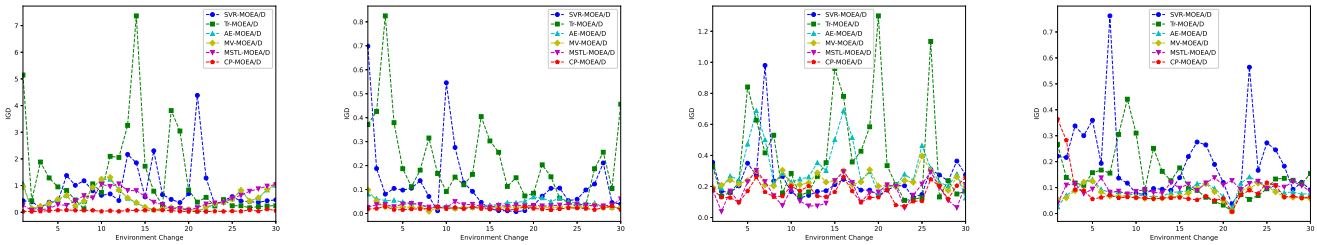
Overall, the proposed *CP-MOEA/D* demonstrated superior performance compared to the other algorithms with respect to the MIGD and MHV metrics, as evident from the aforementioned tabular results. However, it should be noted that *CP-MOEA/D* did not achieve optimal results for almost half of the DF2, DF11 and DF12 instances. In terms of DF2 and DF12, these results are mainly attributed to their static DPOF over time. Since the optima actually remains stationary in the objective space, *CP-MOEA/D* will lose the advantage of dynamic feature learning in solving this particular kind of DMOP. As for DF11, it features the time-varying shrinkage and expansions of the POF segments, which is rare in the other DF instances of the test suite. This results in the absence of corresponding dynamic features in the training samples for the centralized prediction model, and the learning performance of *CP-MOEA/D* is thus weakened.

B. Performance Comparison of Convergence

In addition to comparing solution quality, Fig. 4 further illustrates the convergence curves, showing the average IGD metrics of the searched non-dominated solution set (over 20 runs). The metrics are obtained from the compared six algorithms on 4 representative test instances to evaluate the performance of the learning model-based methods in responding to dynamic changes. In particular, the dynamic IGD values for the compared algorithms under configuration C_1 (i.e., $n_t = 10, \tau_t = 10$), are shown in the figures, the Y-axis displays the average IGD values, while the X-axis represents the change index. It is evident from these curves that any change in the environment of a DMOP leads to an increase in the IGD metric. This suggests that the learning models may not accurately estimate the dynamic tracking.

Overall, during the entire dynamic periods of the studied DF benchmarks, the proposed centralized prediction model-based approach outperforms the compared algorithms in terms of prediction and tracking ability. In particular, the average IGD values of *CP-MOEA/D* (red curves) are basically located at the lowest end of the Y-axis at most of the periods and fluctuate more gently when a problem change occurs. On the test instances such as DF4 and DF5, significant superior predictions were obtained by *CP-MOEA/D* with the centralized prediction model over the other competitors throughout the whole process of problem-solving. This indicates that the proposed *CP-MOEA/D* maintains faster convergence speed and more stable reaction capacity in the changing environment. Nevertheless, similar to the observations from Table II and Table III, due to some unusual dynamic features existing in DF14, such as the changing size and dimension of its DPOF, *CP-MOEA/D* has obtained relatively poorer performance when dynamic begins to occur in the problem. This is mainly attributed to the lack of fusion of relevant features during the training of the centralized model, thus leading to inaccurate predictions. Fortunately, as the model is updated based on more optimized samples of the problem being solved, its generalization performance will continue to improve.

To study the computational overhead of the compared algorithms in solving the DF test problems, this subsection also compares the CPU running time cost based on the implementation environment of AMD Ryzen 7 5800H CPU@ 3.20GHz. Using MOEA/D as the baseline solver, each algorithm runs independently 20 times under the C_3 dynamic configuration (i.e., $n_t = 5$, $\tau_t = 10$) over 30 environmental changes. According to Fig. 5, the efficiency of the comparison in terms of average running times on 14 DF problems is divided into three tiers, and *CP-MOEA/D* lies in the second tier of all the compared algorithms. *CP-MOEA/D* runs in a clearly shorter time than *SVR-MOEA/D* and *Tr-MOEA/D* on all the test cases. These two methods are also based on knowledge learning or transfer models, but they only consider the information of the current



(a) DF4 (b) DF5 (c) DF8 (d) DF14

Fig. 4. Convergence curves of the averaged IGD (over 20 runs) obtained by all the compared algorithms on representative DF benchmarks, under configuration C_1: $n_t = 10, \tau_t = 10$, (Y-axis: IGD value; X-axis: Index of environmental change).

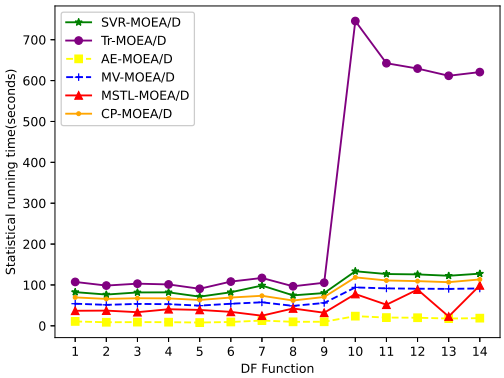


Fig. 5. CPU Running times (in seconds) of compared algorithms.

single problem domain. The running time of *CP-MOEA/D* is slightly longer than *MV-MOEA/D* because *MV-MOEA/D* does not require model learning and training. Among all the compared algorithms, *MSTL-MOEA/D* and *AE-MOEA/D* are in the first tier. *AE-MOEA/D* achieved the lowest computational cost since it only conducts simple linear predictions. However, these two algorithms perform worse than *CP-MOEA/D* in terms of optimization performance. Therefore, *CP-MOEA/D* generally shows not only outstanding dynamic optimization performance but also satisfactory computational efficiency in solving different types of DMOPs.

C. Discussion on Cross-Problem Knowledge Transfer

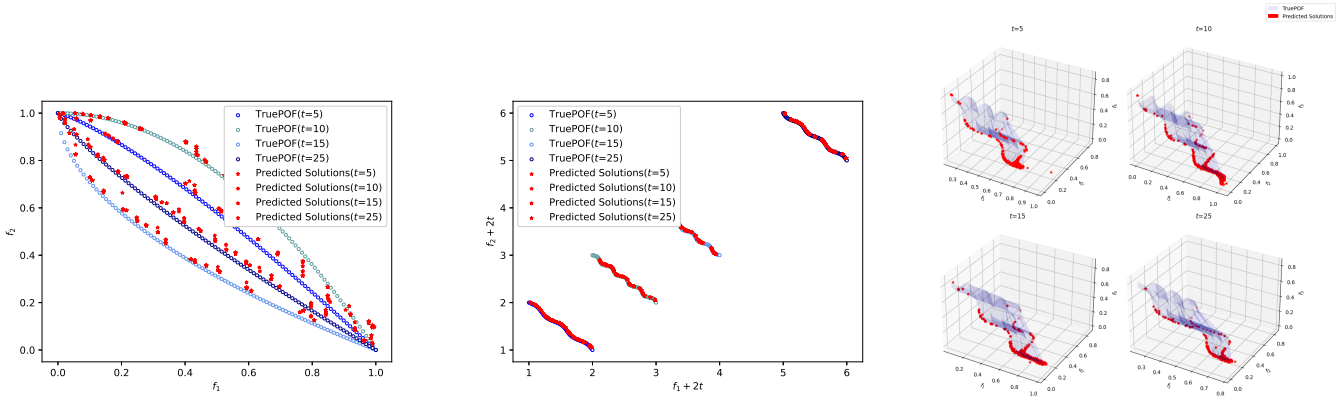
To further delve deeper insights into the superior performance obtained by the proposed centralized prediction with learning across problems to solve DMOPs, as illustrated in Fig. 6, we plot the true POFs and the initial solutions predicted by *CP-MOEA/D* for environments $5, 10, 15$, and 25 under both the fixed dynamic configuration $C1$ in Group $G1$ and the modified shifty dynamic configuration $C6$ in Group $G2$, respectively. As can be observed in Fig. 6, the distribution of solutions obtained through centralized prediction demonstrates its effectiveness in terms of both accuracy and diversity. Notably, in DF5, the algorithm requires nearly no additional iterations to quickly converge to the true optima when dynamics occur. Additionally, to address DF1 in a configuration that is more aligned with real-world applications (i.e., configuration $C6$ with dynamic severity n_t shifting at $t = 10, 20$), the centralized prediction model is developed based on DMOPs previously solved under fixed dynamic configurations in Group $G1$. The obtained predictions in Fig. 6(a) also demonstrate promising tracking ability, indicating the model provides useful guidance when no information about the dynamic patterns of the current-solving problem is available.

The numerical results and analytical illustrations presented above have demonstrated the efficacy of the proposed method in solving DMOPs by learning the correlations across the problem domain. Unlike existing prediction-based methods which can only make predictions when the current problem has undergone sufficient dynamic changes, the proposed centralized learning-assisted prediction is achieved by leveraging the learned common knowledge, i.e., heuristics, dynamic features and search strategies derived from the solved similar tasks, to predict high-quality solutions for the current task.

VI. Conclusion

When an environmental change is detected while solving an encountered DMOP, it is essential for algorithms to respond swiftly. Among existing change response approaches, learning and prediction-based models are particularly effective as they can identify change patterns from past search experiences, thereby directing the evolutionary search towards the dynamic POS or POF as needed. However, these algorithms only focus on the dynamic optimization knowledge obtained within the single problem being solved currently, while the knowledge learning across different DMOPs is still to be exploited.

To fill the gap, in this paper, we have proposed a centralized learning-based prediction model for capturing generalized optimization knowledge in solving DMOPs. The model aims to provide guidance for the evolutionary optimization process, even when dynamic changes occur in the problem. This is in contrast to existing learning and prediction-based approaches, which rely on historical problem data within the same DMOP being processed. In particular, the prediction model consists of two main modules: 1) The shared extractor merges the diverse features of solved problems, and 2) the task-specific classifiers to further exploit the fused dynamic features and identify the category to which the current change belongs. Based on the estimated category-specific selection probabilities, the



(a) DF1-C_6 (b) DF5-C_1 (c) DF14-C_1

Fig. 6. Initial solutions predicted by *CP-MOEA/D* on DF1, DF5 and DF14 under both the fixed (C_1) and the shifty (C_6) dynamic configurations at different experimental changes, respectively.

evolutionary search following a dynamic change can be guided to accurately track the dynamic Pareto optimal set. To assess the overall performance of the proposed CP-DMOEA, comparative experiments were conducted using four state-of-the-art learning and prediction-based baseline algorithms on standard DF test functions with fixed and shifty dynamic configurations. The experimental results reveal that CP-DMOEA consistently outperforms the other algorithms across most test instances, highlighting its effectiveness in incorporating learning across problems into evolutionary dynamic multi-objective optimization.

Moving forward, we plan to delve deeper into the study of solving DMOPs with constraints and real dynamic features via centralized learning-based prediction. Constrained problems are commonly found in natural systems, and so far, we have only included boundary checks in our predictions rather than specific designs for dynamic constraints. Additionally, we will further explore the performance of the proposed CP-DMOEA on DMOPs with a higher number of objectives and apply it to more real-world applications.

REFERENCES

[1] W. Du, W. Zhong, Y. Tang, W. Du, and Y. Jin, "High-dimensional robust multi-objective optimization for order scheduling: A decision variable classification approach," *IEEE transactions on industrial informatics*, vol. 15, no. 1, pp. 293–304, 2018.

[2] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Optimal motion planning in unknown workspaces using integral reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6926–6933, 2022.

[3] J. Fu, C. Zou, M. Zhang, X. Lu, and Y. Li, "Multiobjective dynamic optimization of nonlinear systems with path constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 3, pp. 1530–1542, 2022.

[4] H. Han, L. Zhang, H.-X. Liu, C. Yang, and J. Qiao, "Intelligent optimal control system with flexible objective functions and its applications in wastewater treatment process," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3464–3476, 2019.

[5] D. Gong, M. Rong, N. Hu, Y. Wang, W. Pedrycz, and S. Yang, "A prediction and weak coevolution-based dynamic constrained multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2024.

[6] R. Azzouz, S. Bechikh, and L. Ben Said, "Dynamic multi-objective optimization using evolutionary algorithms: a survey," *Recent advances in evolutionary multi-objective optimization*, pp. 31–70, 2017.

[7] S. Jiang, J. Zou, S. Yang, and X. Yao, "Evolutionary dynamic multi-objective optimisation: A survey," *ACM Computing Surveys*, vol. 55, no. 4, p. 47, nov 2022.

[8] L. T. Bui, Z. Michalewicz, E. Parkinson, and M. B. Abello, "Adaptation in dynamic environments: A case study in mission planning," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 190–209, 2011.

[9] Y. Fang, F. Liu, M. Li, and H. Cui, "Domain generalization-based dynamic multiobjective optimization: A case study on disassembly line balancing," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1851–1865, 2023.

[10] M. Jiang, Z. Wang, S. Guo, X. Gao, and K. C. Tan, "Individual-based transfer learning for dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 10, pp. 4968–4981, 2020.

[11] M. Jiang, Z. Wang, H. Hong, and G. G. Yen, "Knee point-based imbalanced transfer learning for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 117–129, 2020.

[12] X.-F. Liu, X.-X. Xu, Z.-H. Zhan, Y. Fang, and J. Zhang, "Interaction-based prediction for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1881–1895, 2023.

[13] Y. Guo, G. Chen, M. Jiang, D. Gong, and J. Liang, "A knowledge guided transfer strategy for evolutionary dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1750–1764, 2023.

[14] R. Chen, K. Li, and X. Yao, "Dynamic multiobjectives optimization with a changing number of objectives," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, 2017.

[15] K. Zhang, C. Shen, X. Liu, and G. G. Yen, "Multiobjective evolution strategy for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 974–988, 2020.

[16] J. Zhou, J. Zou, S. Yang, G. Ruan, J. Ou, and J. Zheng, "An evolutionary dynamic multi-objective optimization algorithm based on center-point prediction and sub-population autonomous guidance," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 2148–2154.

[17] 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, 2013.

[18] X. Lin, W. Luo, P. Xu, Y. Qiao, and S. Yang, "Popdmmo: A general framework of population-based stochastic search algorithms for dynamic multimodal optimization," *Swarm and Evolutionary Computation*, vol. 68, p. 101011, 2022.

[19] Y. Huang, W. Zhou, Y. Wang, M. Li, L. Feng, and K. C. Tan, "Evolutionary multitasking with centralized learning for large-scale combinatorial multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2023.

[20] Q. Chen, J. Ding, G. G. Yen, S. Yang, and T. Chai, "Multipopulation evolution-based dynamic constrained multiobjective optimization under diverse changing environments," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 3, pp. 763–777, 2024.

[21] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Transactions on evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2016.

[22] G. Ruan, G. Yu, J. Zheng, J. Zou, and S. Yang, "The effect of diversity maintenance on prediction in dynamic multi-objective optimization," *Applied Soft Computing*, vol. 58, pp. 631–647, 2017.

[23] W. Che, J. Zheng, Y. Hu, J. Zou, and S. Yang, "Dynamic constrained multi-objective optimization algorithm based on co-evolution and diversity enhancement," *Swarm and Evolutionary Computation*, vol. 89, p. 101639, 2024.

[24] S. Sahmoud and H. R. Topcuoglu, "A memory-based nsga-ii algorithm for dynamic multi-objective optimization problems," in *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30–April 1, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 296–310.

[25] Q. Zhang, S. Yang, S. Jiang, R. Wang, and X. Li, "Novel prediction strategies for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 260–274, 2020.

[26] J. Li, T. Sun, Q. Lin, M. Jiang, and K. C. Tan, "Reducing negative transfer learning via clustering for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 1102–1116, 2022.

[27] K. Yu, D. Zhang, J. Liang, B. Qu, M. Liu, K. Chen, C. Yue, and L. Wang, "A framework based on historical evolution learning for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 4, pp. 1127–1140, 2024.

[28] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via kalman filter prediction," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2015.

[29] L. Feng, W. Zhou, W. Liu, Y.-S. Ong, and K. C. Tan, "Solving dynamic multiobjective problem via autoencoding evolutionary search," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 2649–2662, 2020.

[30] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, "Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 305–319, 2019.

[31] Y. Hu, J. Zheng, S. Jiang, S. Yang, J. Zou, and R. Wang, "A mahalanobis distance-based approach for dynamic multiobjective optimization with stochastic changes," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 238–251, 2024.

[32] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, "A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization," *IEEE transactions on cybernetics*, vol. 50, no. 12, pp. 5099–5112, 2019.

[33] W. Song, S. Liu, X. Wang, Y. Guo, S. Yang, and Y. Jin, "Learning to guide particle search for dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 54, no. 9, pp. 5529–5542, 2024.

[34] Y. Xu, Y. Song, D. Pi, Y. Chen, S. Qin, X. Zhang, and S. Yang, "A reinforcement learning-based multi-objective optimization in an interval and dynamic environment," *Knowledge-Based Systems*, vol. 280, p. 111019, 2023.

[35] Z. Liang, Y. Zou, S. Zheng, S. Yang, and Z. Zhu, "A feedback-based prediction strategy for dynamic multi-objective evolutionary optimization," *Expert Systems with Applications*, vol. 172, p. 114594, 2021.

[36] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer learning-based dynamic multiobjective optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, 2017.

[37] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, and K. C. Tan, "A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning," *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3417–3428, 2020.

[38] Y. Hu, J. Zou, J. Zheng, S. Jiang, and S. Yang, "A new framework of change response for dynamic multi-objective optimization," *Expert Systems with Applications*, vol. 248, p. 123344, 2024.

[39] Q. Lin, Y. Ye, L. Ma, M. Jiang, and K. C. Tan, "Dynamic multiobjective evolutionary optimization via knowledge transfer and maintenance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 2, pp. 936–949, 2024.

[40] W. Zhou, L. Feng, K. C. Tan, M. Jiang, and Y. Liu, "Evolutionary search with multiview prediction for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 911–925, 2021.

[41] D. Chen, F. Zou, R. Lu, and X. Wang, "A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization," *Swarm and evolutionary computation*, vol. 43, pp. 147–165, 2018.

[42] L. Yan, W. Qi, J. Liang, B. Qu, K. Yu, C. Yue, and X. Chai, "Interindividual correlation and dimension-based dual learning for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1780–1793, 2023.

[43] D. Xu, M. Jiang, W. Hu, S. Li, R. Pan, and G. G. Yen, "An online prediction approach based on incremental support vector machine for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 690–703, 2022.

[44] S. Sahmoud and H. R. Topcuoglu, "Exploiting characterization of dynamism for enhancing dynamic multi-objective evolutionary algorithms," *Applied Soft Computing*, vol. 85, p. 105783, 2019.

[45] K. Deb, U. B. Rao N., and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling," in *Evolutionary Multi-Criterion Optimization*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 803–817.

[46] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[48] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser, and N. Krasnogor, "Benchmark functions for the cec'2018 competition on dynamic multiobjective optimization," Newcastle University, Tech. Rep., 2018.

[49] M. Helbig and A. P. Engelbrecht, "Performance measures for dynamic multi-objective optimisation algorithms," *Information Sciences*, vol. 250, pp. 61–81, 2013.

[50] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on evolutionary computation*, vol. 7, no. 2, pp. 117–132, 2003.

[51] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE transactions on evolutionary computation*, vol. 10, no. 1, pp. 29–38, 2006.

[52] Y. Ye, Q. Lin, L. Ma, K.-C. Wong, M. Gong, and C. A. C. Coello, "Multiple source transfer learning for dynamic multiobjective optimization," *Information Sciences*, vol. 607, pp. 739–757, 2022.

[53] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[54] M. Masum, H. Shahriar, H. Haddad, M. J. H. Faruk, M. Valero, M. A. Khan, M. A. Rahman, M. I. Adnan, A. Cuzzocrea, and F. Wu, "Bayesian hyperparameter optimization for deep neural network-based network intrusion detection," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 5413–5419.

[55] X. Luo, L. O. Oyedele, A. O. Ajayi, O. O. Akinade, J. M. D. Delgado, H. A. Owolabi, and A. Ahmed, "Genetic algorithm-determined deep feedforward neural network architecture for predicting electricity consumption in real buildings," *Energy and AI*, vol. 2, p. 100015, 2020.