Solving Dynamic Multiobjective Problem via Autoencoding Evolutionary Search

Liang Feng, Wei Zhou, Weichen Liu, Yew-Soon Ong, *Fellow, IEEE*, and Kay Chen Tan, *Fellow, IEEE*

***Abstract*—Dynamic multiobjective optimization problem (DMOP) denotes the multiobjective optimization problem, which contains objectives that may vary over time. Due to the widespread applications of DMOP existed in reality, DMOP has attracted much research attention in the last decade. In this article, we propose to solve DMOPs via an autoencoding evolutionary search. In particular, for tracking the dynamic changes of a given DMOP, an autoencoder is derived to predict the moving of the Pareto-optimal solutions based on the nondominated solutions obtained before the dynamic occurs. This autoencoder can be easily integrated into the existing multiobjective evolutionary algorithms (EAs), for example, NSGA-II, MOEA/D, etc., for solving DMOP. In contrast to the existing approaches, the proposed prediction method holds a closed-form solution, which thus will not bring much computational burden in the iterative evolutionary search process. Furthermore, the proposed prediction of dynamic change is automatically learned from the nondominated solutions found along the dynamic optimization process, which could provide more accurate Pareto-optimal solution prediction. To investigate the performance of the proposed autoencoding evolutionary search for solving DMOP, comprehensive empirical studies have been conducted by comparing three state-of-the-art prediction-based dynamic multiobjective EAs. The results obtained on the commonly used DMOP benchmarks confirmed the efficacy of the proposed method.**

***Index Terms*—Autoencoding, dynamic multiobjective optimization, evolutionary optimization, knowledge transfer.**

Manuscript received June 30, 2019; revised November 11, 2019 and June 17, 2020; accepted August 5, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61876025 and Grant 61876162; in part by the Venture and Innovation Support Program for Chongqing Overseas Returnees under Grant cx2018044 and Grant cx2019020; in part by the City University of Hong Kong Research Fund under Grant 9610397; in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20180307123637294; and in part by the Research Grants Council of Hong Kong SAR under Grant CityU11202418 and Grant CityU11209219. This article was recommended by Associate Editor G. G. Yen. *(Corresponding author: Liang Feng.)*

Liang Feng and Wei Zhou are with the Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, Chongqing 400044, China, and also with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: liangf@cqu.edu.cn; jerryzhou@cqu.edu.cn).

Weichen Liu and Yew-Soon Ong are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: liu@ntu.edu.sg; asysong@ntu.edu.sg).

Kay Chen Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the City University of Hong Kong Shenzhen Research Institute, Shenzhen, China (e-mail: kaytan@cityu.edu.hk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.3017017

I. Introduction

A MULTIOBJECTIVE optimization problem (MOP) involves more than one objective function to be optimized simultaneously. It arises in many real-world applications, such as engineering, economics, and logistics, where the corresponding optimal decisions need to be made in the presence of tradeoffs between two or more conflicting objectives [1], [2]. For example, developing a vehicle routing plan might involve minimizing the number of vehicles while maximizing the number of customers to be served. In contrast to single-objective optimization, which tries to find the single optimal solution, MOP is more challenging and aims to obtain the optimal solution set which is called the Pareto-optimal solutions (POSs), comprising a number of solutions that fair equivalently according to the Pareto dominance concept [3], [4].

To solve the MOPs, evolutionary algorithms (EAs), which are inspired by the biological evolution and Darwinian evolutionary principles, have demonstrated strong optimization capability of searching the POS [5]. In the literature, many multiobjective EAs (MOEAs) have been developed in the past decades [6]–[9]. However, in reality, due to the uncertainty that exists in nature, most MOPs are dynamic MOPs (DMOPs), in which the problem properties, such as objective functions, function constraints, etc., are varying over time. The POS of a DMOP found at a particular time instance thus may not be optimal at another. To tackle DMOP, many research works have been proposed in the literature by extending the existing MOEAs with additional designs to track the moving POS in the changing environment [10]–[17]. In particular, according to the recent surveys [18], [19], the existing dynamic MOEAs (DMOEAs) can be generally categorized as follows: 1) diversity-based approaches (e.g., D-NSGA-II [20]), which either introduce diversity when dynamic occurs or maintain high diversity throughout the search process [21]–[25]; 2) prediction-based approaches (e.g., population prediction strategy (PPS) [26], MOEA/D-KF [27], and Tr-DMOEA [14]), which propose to predict the dynamic changes based on the latent patterns learned from the search experiences [16], [28], [29]; 3) memory-based approaches (e.g., d-COEA [30] and MS-MOEA [31]), that mainly reuse previous optimal solutions stored to quickly respond to the new environmental change [32]–[36]; and 4) multipopulation approaches (e.g., dynamic PSFGA [37]) which maintain multiple subpopulations for search concurrently [38]. The fourth category can be treated as a combination of the diversity-based approach and memory-based approach. For more reviews of the DMOEAs, interested readers can refer to [18], [19], and [39].

As changes in dynamic environments may exhibit some patterns that are predictable, among the existing DMOEAs for solving DMOP, prediction-based approaches have illustrated superior optimization performance in terms of both search speed and solution quality in contrast to other categories of DMOEAs [14], [26], [27], [40], [41]. In recent years, an increasing number of research efforts have been proposed to explore the advanced prediction methods for tracking dynamic changes, toward efficient and effective problem solving of DMOPs. For instance, Zhou *et al.* [26] presented a PPS, to predict the entire population of MOEA. The proposed PPS mainly contains two parts, that is, center point prediction and manifold prediction. Muruganantham *et al.* [27] proposed a new DMOEA based on Kalman filter (KF) predictions. The predictions help guide the search toward the changed optima; thereby, the DMOEA can quickly track the moving optima in the decision space. Furthermore, Jiang *et al.* [14] introduced the integration of transfer learning and MOEAs for solving DMOP. The authors employed the transfer component analysis (TCA) for prediction in the objective space and verified the effectiveness of the proposed approach with several existing MOEAs. Zou *et al.* [40] proposed a prediction strategy based on both center points and knee points. The proposed algorithm mainly contains three mechanisms, which are the center points-based prediction, the knee point-based prediction, and the adaptive diversity maintenance. More recently, Rong *et al.* [41] presented a multidirectional prediction strategy to enhance the performance of MOEAs in solving DMOP. Instead of using the center point, a number of representative groups of the POS were proposed to predict the moving of optima. Cao *et al.* [42] presented a difference model to predict the motion of POS based on the historical centroid locations.

Autoencoding evolutionary search is a recently proposed search paradigm with learning from past search experiences [43]. In [43], a single-layer denoising autoencoder is derived to build the connection across problem domains by finding solution mapping from one domain to the other. The autoencoder thus provides a possibility of learning the relationship between solutions across domains. In the context of DMOP, from the discussion of the existing prediction-based DMOEAs above, we can see that the problems at two consecutive time instances of a given DMOP usually share a great similarity, and the solutions (e.g., center points in [26] and representative points in [41]) of the DMOP at one time instance are thus able to help the prediction of the moving optima of the next time instance. Keeping these in mind, in this article, we propose a new prediction strategy by extending the autoencoding evolutionary search for solving DMOP. In contrast to the existing prediction approaches, instead of using the deterministic data points, such as center point, knee point, etc., the proposed method tracks the moving of optima by learning the mapping from the nondominated solutions found in two consecutive time instances at which the dynamic occurs. The main contributions of this work can be summarized as follows.

1) To track the moving of POS in DMOP, an extremely efficient autoencoder which holds a closed-form solution has been proposed.
2) As the proposed direction prediction of moving POS is automatically learned from the nondominated solutions along the dynamic optimization process, it could provide more diverse and accurate POS prediction than the existing methods using only deterministic data points, such as center point, knee point, etc.
3) The proposed method is optimizer independent, it thus can be easily integrated into any of the existing static MOEAs for tackling DMOPs.
4) To verify the efficacy of the proposed method, comprehensive empirical studies have been conducted on the commonly used DMOP benchmarks that possess various dynamics. Three state-of-the-art prediction-based DMOEAs using different evolutionary optimizers are considered as the baselines for comparison. The obtained results confirmed the efficiency and effectiveness of the proposed autoencoding evolutionary search for solving DMOPs.

The remainder of this article is organized as follows. The definition of the DMOP studied in this article, and the brief introduction of the autoencoding evolutionary search paradigm proposed in [43], are presented in Section II. Section III gives the details of the proposed method for solving DMOPs. Furthermore, Section IV provides comprehensive empirical studies on commonly used DMOP benchmarks, against three state-of-the-art prediction-based MOEAs. Finally, we draw the concluding remarks of this article in Section V.

II. Preliminary

In this section, we first give the definition of the DMOP investigated here. Next, the brief introduction of the autoencoding evolutionary search paradigm proposed in [43], which inspired the proposed method in this article for solving DMOP, is also presented.

A. Dynamic Multiobjective Optimization Problem

In this article, we consider the DMOP as a time-variant MOP. In particular, a minimization problem is considered here. Mathematically, a DMOP can be defined as follows [5], [14], [18]:

$$\begin{aligned} \min \quad & F(\mathbf{x}, t) = \left[f_1(\mathbf{x}, t), \ldots, f_m(\mathbf{x}, t)\right]^T \\ \text{s.t.} \quad & G(\mathbf{x}, t) \leq 0,\ H(\mathbf{x}, t) = 0,\ \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where Ω is the decision (variable) space and $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector of decision variables. f_i represents the ith objective function, while m gives the total number of objectives to be minimized. $F(\mathbf{x}, t)$ is the set of objective values with respect to time or other dynamics of the DMOP, which is represented by t. The functions G and H represent the set of inequality and equality constraints, respectively. L_i and U_i are the lower and upper bounds of x_i, respectively.

At time t, a decision vector $\mathbf{x}_1$ is Pareto dominated by another vector $\mathbf{x}_2$, denoted by $\mathbf{x}_1 \prec_t \mathbf{x}_2$, if and only if

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

Based on the concept of dynamic Pareto dominance, the definitions of dynamic Pareto-optimal set (DPOS) and dynamic Pareto-optimal front (DPOF) are given as follows.

Definition 1 (DPOS): The POS at time t, denoted as $\mathrm{DPOS}(t)^*$, is the set of all Pareto-optimal solutions with respect to the decision space such that

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

Definition 2 (DPOF): The POF at time t, denoted as $\mathrm{DPOF}(t)^*$, is the corresponding objective vectors of the $\mathrm{DPOS}(t)^*$ such that

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

The target of a DMOEA is thus to evolve the population and trace the moving DPOS or DPOF as fast as possible when dynamic occurs.

B. Autoencoding Evolutionary Search

In the literature, an autoencoder is the basic building block of deep learning networks that attempts to reproduce its input, that is, the target output is equal to the input itself [44], [45]. In the last decade, it has attracted great research attention, and many variants of autoencoders have been proposed to improve machine-learning performance. For instance, Vincent *et al.* [46] proposed to extract and compose robust features via the denoising autoencoder, while Rifai *et al.* [47] introduced the training of deterministic autoencoder which incorporates a well-chosen penalty term to the classical autoencoding reconstruction cost function. Moreover, Sun *et al.* presented an unsymmetrical autoencoder and an explicitly guided autoencoder for learning good and meaningful feature in [48] and [49], respectively.

Recently, beside learning high-quality feature in machine learning, autoencoder has been derived to build the connection between two independent optimization domains. In particular, in [43], toward the improved optimization process on unseen problems, the autoencoding evolutionary search has been proposed to learn from past search experiences. Specifically, by devising a single-layer denoising autoencoder and treating the solved problem as the corrupted version of the present problem, the optimized solutions of past solved problems can be transferred, in the form of solutions for the present optimization problem, to speed-up the corresponding search process. Specifically, according to [43], let $\mathbf{P} \in \mathbb{R}^{d\times N}$ and $\mathbf{Q} \in \mathbb{R}^{d\times N}$ represent the set of solutions of two different optimization problems $\mathbf{OP}_1$ and $\mathbf{OP}_2$, respectively, that is, $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ and $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_N\}$, where N denotes the number of solutions in each set and d is the problem dimension. The connection $\mathbf{M} \in \mathbb{R}^{d\times d}$ from $\mathbf{OP}_1$ to $\mathbf{OP}_2$ can be naturally built through a denoising autoencoder by using $\mathbf{P}$ as the input and $\mathbf{Q}$ as the output accordingly, which is given by

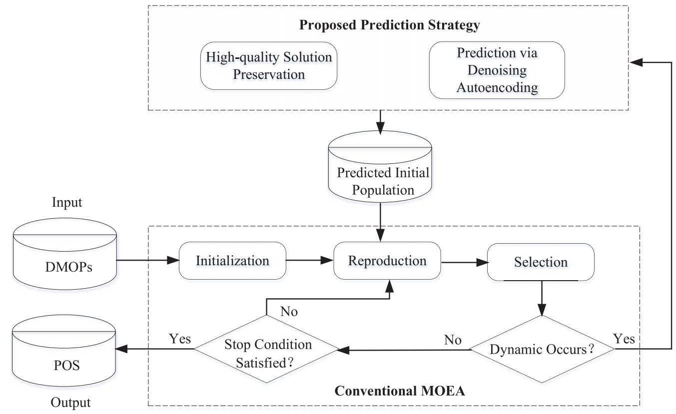

Fig. 1. Outline of the proposed autoencoding evolutionary search for solving DMOPs.

$$L(\mathbf{M}) = \frac{1}{2N} \sum_{i=1}^{N} ||\mathbf{q}_i - \mathbf{M}\mathbf{p}_i||^2. \tag{5}$$

According to [43], (5) holds a closed-form solution, which is given by

$$\mathbf{M} = \left(\mathbf{Q}\mathbf{P}^{\mathrm{T}}\right)\left(\mathbf{P}\mathbf{P}^{\mathrm{T}}\right)^{-1} \tag{6}$$

where T is the transpose operation of a matrix. The knowledge transfer from the problem domain $\mathbf{OP}_1$ to $\mathbf{OP}_2$ is simply realized by the multiplication of $\mathbf{M}$ and the optimized solutions of $\mathbf{OP}_1$.

As can be observed in (5), $\mathbf{M}$ serves to map solutions in $\mathbf{OP}_1$ domain to the solutions in the domain of $\mathbf{OP}_2$. In the context of DMOP, it is straightforward to see that, if we treat the DMOPs before and after the dynamic occurs, as $\mathbf{OP}_1$ and $\mathbf{OP}_2$, the learned $\mathbf{M}$ essentially captures the moving directions of the DMOP. In contrast to the existing works, which deterministically use the differences between center points for preserving the moving direction of DMOP, $\mathbf{M}$ could lead to more diverse and accurate prediction of the searching direction for solving the DMOP, since it is automatically learned from the MO solutions found before and after the dynamic occurs. Inspired by this, here we propose to extend the idea of autoencoding in the decision space to build the knowledge transfer mapping for solving DMOPs, which will be detailed in the next section.

III. Proposed Method

In this section, the details of the proposed autoencoding evolutionary search for solving DMOPs are presented. In particular, as outlined in Fig. 1, by employing a conventional MOEA as the basic optimization solver, the search process starts as routine, following the procedures, such as initialization, reproduction, and selection. However, an additional detection operator is designed to trigger the proposed prediction strategy for generating a new population to predict the moving POS. The proposed prediction strategy consists of two components, which are the *prediction via denoising autoencoding* and *high-quality solution preservation*. The

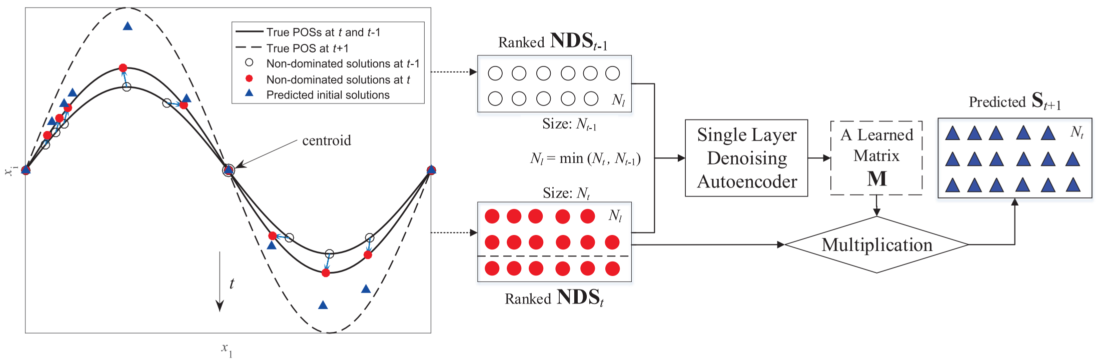

Fig. 2. Illustration of the proposed POS prediction via denoising autoencoding.

entire evolutionary search process will be iteratively performed until the predefined stopping criteria are satisfied.

In what follows, the details of the dynamic detection operator, the denoising autoencoding operator, and the genetic preservation operator are presented.

A. Dynamic Detection

The process of dynamic detection enables an MOEA in the dynamic problem environment to respond with necessary operations for maintaining the optimization performance. In the literature, according to a survey [50], the existing change detection methods can be generally categorized into two groups, that is: 1) population-based detection and 2) sensor-based detection. In particular, the population-based detection methods use the fitness evaluations of the entire population of an EA, while the sensor-based detection approaches consider additional measurement of the fitness landscape on certain predefined points.

In our proposed method, following the successes achieved in previous DMOEAs [11], [40], a simple and effective sensor-based detection method is employed. Particularly, we randomly select 5% individuals as detectors in the population and archive their objective values. At the start of each generation, the detectors will be re-evaluated, and a mismatch in the objective values then suggests that a change of the DMOP has occurred.[1]

B. Proposed Population Generation When Dynamic Occurs

When a dynamic change of the given DMOP has been detected, we propose to regenerate the population with the aim of locating the new POS quickly. In particular, the proposed population generation approach consists of two parts, each generating half of the new population. The first part is *prediction via denoising autoencoding*, which aims to predict the new POS by tracing the moving direction of POS. The other part is *high-quality solution preservation*, which is to preserve the nondominated solutions found along the entire evolutionary search process.

1) Prediction via Denoising Autoencoding: To predict the new POS after the dynamic change, one of the key challenges is to predict the moving direction of the POS based on the detected change of the optimization problem. In the literature, a popular approach is to use the difference between center points of the POS found in former time windows [or dynamic environment $f(\mathbf{x}, t)$] to represent the moving direction of POS. However, it is worth noting that this difference can only provide a single deterministic direction for prediction, which could not be the true moving direction of all the solutions in the POS. An inappropriate prediction of the moving direction may even hinder the convergence to the POS. Keeping this in mind, instead of predicting a single direction, here we propose to predict multiple moving directions of POS by considering the moving of each nondominated solution found in former time windows. The proposed method thus has a higher probability to discover the true moving direction of POS, in contrast to the traditional center points-based prediction.

In particular, suppose a series of nondominated solutions obtained in the previous time windows is denoted as $\mathbf{NDS}_1$, $\mathbf{NDS}_2, \ldots, \mathbf{NDS}_{t-1}$, $\mathbf{NDS}_t$, in this article, we propose to use the nondominated solutions in $\mathbf{NDS}_{t-1}$ and $\mathbf{NDS}_t$ to learn the moving directions of POS solutions from time window t to the next time window $t+1$, which is detailed in Algorithm 1 and illustrated in Fig. 2. Specifically, suppose N_{t-1} and N_t denote the number of nondominated solutions in $\mathbf{NDS}_{t-1}$ and $\mathbf{NDS}_t$, respectively, we first rank the $\mathbf{NDS}_{t-1}$ and $\mathbf{NDS}_t$ solutions in ascending orders independently, with respect to the crowd distance [6]. Next, let $N_l = \min(N_{t-1}, N_t)$, by using the best N_l solutions in the ranked $\mathbf{NDS}_{t-1}$, and another best N_l solutions in the ranked $\mathbf{NDS}_t$, as the input $\mathbf{P}$ and output $\mathbf{Q}$ of the denoising autoencoder discussed in Section II-B, respectively, the moving directions of POS from time window $t-1$ to t can be modeled by the learned matrix $\mathbf{M}$ via (6). As can be observed, since the learned $\mathbf{M}$ tries to map each of the N_l solutions in $\mathbf{NDS}_{t-1}$ to different solutions in $\mathbf{NDS}_t$ based on the obtained ranks, it provides various solution-specific directions for predicting the moving POS. Furthermore, the learned $\mathbf{M}$ is used to predict the POS solutions for time window $t+1$ via

$$\mathbf{S}_{t+1} = \mathbf{S}_t \times \mathbf{M} \tag{7}$$

where $\mathbf{S}_{t+1}$ and $\mathbf{S}_t$ are $N_t \times d$ matrices (d is the dimension of the given DMOP, and each row of the matrices denotes one solution.), representing the predicted POS solutions and the nondominated solutions of $\mathbf{NDS}_t$, respectively. Finally, let *NP*

[1]Without loss of generality, another dynamic detection approach can also be applied.

TABLE I
DETAILED PROPERTY OF THE DMOPS

Name	# of Dim.	# of Obj.	Properties
FDA4	12	3	TYPE I Non-convex POF
FDA5	12	3	TYPE II Non-convex POF
FDA5$_{iso}$	12	3	TYPE II Isolated, non-convex POF
FDA5$_{dec}$	12	3	TYPE II Deceptive, non-convex POF
DIMP2	10	2	TYPE I Convex POF, different change rates of decision variables
dMOP2	10	2	TYPE II POF changes from convex to concave
dMOP2$_{iso}$	10	2	TYPE II Isolated POF, changes from convex to concave
dMOP2$_{dec}$	10	2	TYPE II Deceptive POF, changes from convex to concave
dMOP3	10	2	TYPE I Convex POF, spread of solutions changes over time
HE2	30	2	TYPE III Discontinuous POF

Algorithm 1: Pseudocode of the Proposed POS prediction via Denoising Autoencoding

Input : $\mathbf{NDS}_{t-1}$ and $\mathbf{NDS}_t$: POS solutions obtained in time window $t-1$ and t, respectively. N_t and N_{t-1}: the number of solutions in $\mathbf{NDS}_t$ and $\mathbf{NDS}_{t-1}$, respectively.

Output: $\mathbf{S}_{t+1}$: Predicted POS solutions for time window $t+1$.

Begin

Rank solutions in $\mathbf{NDS}_{t-1}$ and $\mathbf{NDS}_t$, separately, using crowd distance.

Configure the input $\mathbf{P}$ and output $\mathbf{Q}$ of the denoising autoencoder in section II-B, as the best N_l solutions in the ranked $\mathbf{NDS}_{t-1}$ and another best N_l solutions in the ranked $\mathbf{NDS}_t$, respectively, where $N_l = \min(N_t, N_{t-1})$.

Obtain the matrix $\mathbf{M}$ representing the predicted moving directions of POS solutions via Eq. 6.

Calculate the predicted POS solutions $\mathbf{S}_{t+1}$ for time window $t+1$ using Eq. 7.

End

denotes the population size of the evolutionary solver, if $N_t \geq (NP/2)$, all the solutions in $\mathbf{S}_{t+1}$ will be ranked using crowd distance, and the best $(NP/2)$ solutions (i.e., nondominated solution plus the top-ranked solutions) are then selected to serve as the initial solutions of the evolutionary search for time window $t+1$. Otherwise, all solutions in $\mathbf{S}_{t+1}$ will be used as the initial solutions.

2) High-Quality Solution Preservation: For the other half of the predicted new population, we propose to randomly select $(NP/2)$ solutions from $\mathbf{NDS}_t$. This is to, first, preserve the high-quality solutions found along the evolutionary search process, and second, to maintain the diversity of the population for further exploration of the evolutionary search.

Last but not least, if the predicted solutions in Section III-B1 plus the preserved solutions from $\mathbf{NDS}_t$ are less than the population size NP, randomly generated solutions will be used to fill the population.

IV. EMPIRICAL STUDY

In this section, to evaluate the efficacy of the proposed method for solving DMOPs, comprehensive empirical studies on commonly used DMOP benchmarks against recently proposed DMOEAs are presented.

TABLE II
CONFIGURATIONS OF THE DYNAMIC CHANGES ON THE DMOPS

C	n_t	τ_t
C_1	10	5
C_2	10	10
C_3	10	25
C_4	10	50
C_5	1	10
C_6	1	50
C_7	20	10
C_8	20	50

A. Experimental Setup

1) Test Instances: In this study, ten DMOPs from the IEEE CEC 2015 dynamic benchmark problems [51]–[53] are considered as the test problems. These benchmark problems possess diverse properties, such as the number of objectives, number of variable dimensions, and POF shape, and belong to different types of DMOP, which are categorized by Farina *et al.* [54]. In particular, the details of the considered DMOPs are summarized in Table I. In the table, type I implies the DMOPs that have varying POS, type II means that both the POS and POF of the DMOP will change over time, and type III denotes the DMOPs having dynamic POF. Furthermore, in each of these DMOPs, a time index t is usually defined to control the severity and frequency of environmental change [51], [54], which is given by

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

where τ is the generation index, and $\lfloor\cdot\rfloor$ denotes the floor operator. A smaller value of n_t means larger dynamic change, while a smaller value of τ_t leads to a more frequent occurrence of changes in the DMOP. In this article, toward a comprehensive empirical study, according to [51], eight different configurations of dynamic changes are investigated on all the ten DMOPs, which are summarized in Table II.

2) Performance Metric: In the literature, there are several metrics that have been proposed to evaluate the performance of DMOEAs [14], [26], [55], including the IGD metric, mean inverted generational distance (MIGD) metric, HVD metric, etc. To investigate different aspects of a DMOEA in solving DMOPs, such as distribution, convergence, and closeness, of the obtained solutions to the true POF, here we employ the popular IGD metric and two of its variants as the performance metrics.

In particular, the IGD metric [26] is a performance indicator that quantifies the proximity and diversity of a multiobjective optimization algorithm, which is given by

$$\mathrm{IGD}\left(P^{t*}, P^{t}\right) = \frac{\sum_{v \in P^{t*}} \min_{v \in P^{t}} \|v^{*} - v\|}{|P^{t*}|} \tag{9}$$

where P^{t*} is the set of uniformly distributed points in the true POF at time t and P^{t} represents the POF obtained by the algorithm, respectively. The IGD metric can measure both the diversity and convergence of the search, and a lower value of IGD indicates a superior optimization performance.

The second metric is the MIGD metric [11], which is a variant of the IGD metric for evaluating DMOEAs. It calculates the average of the IGD values in some time steps over a run, which is defined as

$$\mathrm{MIGD}\left(P^{t*}, P^{t}\right) = \frac{1}{|T|} \sum_{t \in T} \mathrm{IGD}\left(P^{t*}, P^{t}\right) \tag{10}$$

where T is a set of discrete-time instances in a run, and $|T|$ is the cardinality of T.

Furthermore, to investigate the robustness of a DMOEA in different dynamic environments, the DMIGD defined in [14] is considered as the third performance metric here, which is defined as follows:

$$\mathrm{DMIGD}\left(P^{t*}, P^{t}, C_i\right) = \frac{1}{|E|} \sum_{C_i \in E} \mathrm{MIGD}\left(P^{t*}, P^{t}, C_i\right) \tag{11}$$

where $|E|$ is the number of different environments, and C_i denotes the particular dynamic configuration of a DMOP. In this study, as we have eight number of dynamic configurations for each DMOP (see Table II), $|E|$ is then set as 8.

3) Compared Algorithms: To confirm the efficacy of the proposed prediction strategy for solving DMOPs, three state-of-the-art prediction-based DMOEAs are considered here as the baseline algorithms for comparison. As these baselines consider different MOEAs for solving DMOP, the performance of the proposed method when integrated into different MOEAs will be investigated. In particular, the first one is a transfer learning-based NSGA-II, namely, *Tr-NSGA-II*, proposed by Jiang *et al.* [14], while the second is a multidirectional prediction approach under the framework of particle swarm optimization, called *MDP-MOPSO*, proposed by Rong *et al.* [41]. The third baseline is the MOEA/D equipped with a difference model-based prediction for solving DMOPs, that is, *DM-MOEA/D*, proposed by Cao *et al.* [42]. For a fair comparison, we integrate our proposed autoencoding-based prediction with NSGA-II (labeled as *AE-NSGA-II*), MOPSO (labeled as *AE-MOPSO*), and MOEA/D (labeled as *AE-MOEA/D*), and compare these integrations against the baseline algorithm possessing the same evolutionary solver, respectively.

Next, according to [14], for each DMOP studied, all the algorithms are supposed to stop after 20 environmental changes. The population size of all the compared algorithms is set as 200, and the POS size is set as 200 for DMOEAs using NSGA-II and MOEA/D solver, and 100 for the MOPSO-based DMOEAs, respectively. Other configurations of evolutionary operators and parameters are kept consistent with the settings in [14], [41], and [42] for NSGA-II, MOPSO, and MOEA/D, respectively.

B. Results and Discussion

In this section, we analyze the comparison results first from the perspective of solution quality, which is followed by the discussion on the convergence speed of the compared algorithms when dynamic occurs. The deeper insights of the obtained results are also presented and discussed in this section.

1) Solution Quality: Tables III and IV summarize the averaged MIGD value and the standard deviation obtained by *AE-MOEA/D*, *DM-MOEA/D*, *AE-NSGA-II*, *Tr-NSGA-II*, *AE-MOPSO*, and *MDP-MOPSO*, over 20 independent runs on the two-objective and three-objective DMOPs with different configurations of dynamic changes, respectively. In the tables, superior performances are highlighted in bold. Further, the Wilcoxon rank-sum test with 95% confidence level is conducted on the experimental results. "+," "−," and "≈" denote the proposed autoencoding approach is statistically significantly better, worse, and similar to the compared dynamic multiobjective method which shares the same multiobjective optimizer, respectively. The column "Winner" gives the algorithm that obtained the best averaged MIGD value on the solved DMOP instance.

As can be observed in the tables, each of the multiobjective optimizer equipped with the proposed autoencoding approach (i.e., *AE-MOEA/D*, *AE-NSGA-II*, and *AE-MOPSO*) achieved superior performance in terms of the MIGD value against its counterpart (i.e., *DM-MOEA/D*, *Tr-NSGA-II*, and *MDP-MOPSO*), on most of the DMOPs under different dynamic configurations. In particular, on DMOPs, such as DIMP2, $\mathrm{dMOP2}_{\mathrm{dec}}$, and FDA4, *AE-MOEA/D* obtained fitter MIGD values over *DM-MOEA/D*, under almost all the eight configurations of dynamic changes. Similar results can also be observed between *AE-NSGA-II* and *Tr-NSGA-II*, on DMOPs, such as dMOP2, dMOP3, and FDA5, and *AE-MOPSO* versus *MDP-MOPSO* on DMOP $\mathrm{dMOP2}_{iso}$, FDA4, and $\mathrm{FDA5}_{iso}$. Furthermore, on totally 80 number of DMOP instances, *AE-MOEA/D*, *AE-NSGA-II*, and *AE-MOPSO* obtained superior or competitive MIGD values over *DM-MOEA/D*, *Tr-NSGA-II*, and *MDP-MOPSO*, on 67, 68, and 72 DMOP instances, respectively. However, it has also been observed that the *AE-NSGA-II* lose to *Tr-NSGA-II* on the DIMP2 problem. This is because DIMP2 possesses different change rates of decision variables (see Table I), which may result in highly nonlinear changing of POS, and thus complex and expensive design of

TABLE III
MEAN AND STANDARD DEVIATIONS OF MIGD VALUES OBTAINED BY THE COMPARED ALGORITHM ON DMOPS POSSESSING TWO OBJECTIVES (SUPERIOR PERFORMANCES ARE HIGHLIGHTED IN BOLD. "+," "−," AND "≈" DENOTE THE PROPOSED AUTOENCODING APPROACH IS STATISTICALLY SIGNIFICANT BETTER, WORSE, AND SIMILAR TO THE COMPARED DYNAMIC MULTIOBJECTIVE METHOD THAT SHARES THE SAME MULTIOBJECTIVE OPTIMIZER, RESPECTIVELY.)

Problems	(n_t,τ_t)	*AE-MOEA/D*	*DM-MOEA/D*	*AE-NSGA-II*	*Tr-NSGA-II*	*AE-MOPSO*	*MDP-MOPSO*	Winner
DIMP2	(10,5)	**1.1085E+00(2.3177E-01)**+	2.8401E+00(9.9289E-01)	1.0073E+01(7.5581E-01)–	**6.0887E+00(1.9144E-01)**	**8.1666E+00(1.1284E+00)**≈	8.1901E+00(5.6334E-01)	*AE-MOEA/D*
	(10,10)	**3.9039E-01(9.8162E-02)**+	1.4335E+00(4.7169E-01)	8.2725E+00(6.5418E-01)–	**4.6109E+00(2.8795E-01)**	**6.6393E+00(1.3869E+00)**+	7.9140E+00(4.1731E-01)	*AE-MOEA/D*
	(10,25)	**1.0375E-01(3.1640E-02)**+	4.0169E-01(2.0366E-01)	6.1152E+00(7.8692E-01)–	**3.0603E+00(1.3130E-01)**	**5.8501E+00(1.5769E+00)**+	7.2094E+00(7.1001E-01)	*AE-MOEA/D*
	(10,50)	**2.2711E-02(1.0419E-02)**+	5.1386E-02(3.6327E-02)	4.8560E+00(6.4317E-01)–	**2.5551E+00(1.7218E-01)**	**4.6227E+00(1.8564E+00)**+	6.0290E+00(8.4685E-01)	*AE-MOEA/D*
	(1,10)	**1.5923E+00(2.1878E-01)**+	2.4342E+00(4.4233E-01)	6.2338E+00(1.4538E-01)–	**4.5100E+00(1.7778E-01)**	1.1270E+01(2.3275E+00)–	**7.8304E+00(5.7562E-01)**	*AE-MOEA/D*
	(1,50)	**1.7348E-01(3.0875E-02)**+	5.0714E-01(7.0571E-02)	5.3630E+00(6.8623E-01)–	**2.4926E+00(1.7078E-01)**	6.8976E+00(2.0607E+00)≈	**5.7871E+00(5.1023E-01)**	*AE-MOEA/D*
	(20,10)	**2.9578E-01(1.0676E-01)**+	7.0668E-01(1.9285E-01)	5.6157E+00(2.1754E-01)–	**4.5148E+00(2.6129E-01)**	**7.7726E+00(3.0004E+00)**≈	7.8018E+00(4.2362E-01)	*AE-MOEA/D*
	(20,50)	**2.0074E-02(8.3096E-03)**+	6.1217E-02(3.9834E-02)	4.5679E+00(5.7541E-01)–	**2.5299E+00(3.1434E-01)**	**2.3931E+00(1.2048E+00)**+	5.8498E+00(5.1102E-01)	*AE-MOEA/D*
dMOP2	(10,5)	1.8078E-01(5.7082E-02)–	**1.0625E-01(3.6765E-02)**	**5.8146E-01(2.3385E-01)**+	7.8782E-01(1.7176E-01)	**1.3489E-01(7.6097E-02)**≈	1.3557E-01(2.3177E-02)	*DM-MOEA/D*
	(10,10)	4.0436E-02(2.1337E-02)≈	**2.9431E-02(2.0063E-02)**	**2.5563E-01(9.4797E-02)**+	4.9871E-01(9.5338E-02)	**1.1214E-01(7.6041E-02)**≈	1.2334E-01(4.1788E-02)	*DM-MOEA/D*
	(10,25)	**5.5086E-03(8.9199E-04)**≈	6.1615E-03(6.8380E-04)	**1.0159E-01(4.8839E-02)**+	1.7490E-01(3.6485E-02)	**6.0134E-02(2.8268E-02)**+	1.0080E-01(3.8011E-02)	*AE-MOEA/D*
	(10,50)	**4.3344E-03(7.2343E-04)**≈	4.4402E-03(2.0882E-05)	**3.3852E-02(1.6037E-02)**+	4.8541E-02(1.2273E-02)	**5.1478E-02(3.3088E-02)**≈	5.3377E-02(2.6976E-02)	*AE-MOEA/D*
	(1,10)	**1.8304E+01(2.9491E-02)**+	1.8562E+01(8.7886E-02)	**1.6247E-02(2.7019E-04)**+	7.1903E+00(2.3845E+00)	**1.8842E+01(6.3398E-01)**+	2.0982E+01(7.1086E-01)	*AE-NSGA-II*
	(1,50)	1.8011E+01(3.3387E-03)–	**1.8008E+01(1.3367E-05)**	**1.6600E+00(1.7975E+00)**≈	3.3618E+00(3.9362E+00)	**1.8756E+01(3.8619E-01)**+	1.9798E+01(1.0967E+00)	*AE-NSGA-II*
	(20,10)	**1.3373E-02(7.3001E-03)**≈	1.8315E-02(6.4170E-03)	**2.0886E-01(7.2181E-02)**+	5.7934E-01(1.0092E-01)	3.9443E-01(1.4774E-01)–	**7.4374E-02(2.7308E-02)**	*AE-MOEA/D*
	(20,50)	**4.0679E-03(3.2236E-04)**+	4.1175E-03(1.5683E-05)	**3.5441E-02(1.3433E-02)**+	5.4736E-02(1.3859E-02)	1.4248E-01(9.3420E-02)–	**5.6198E-02(2.3956E-02)**	*AE-MOEA/D*
dMOP2_{iso}	(10,5)	**4.2662E-03(3.3513E-04)**≈	4.3603E-03(5.7075E-05)	3.0586E-03(5.3629E-04)–	**2.6454E-03(6.4218E-05)**	**8.4214E-03(1.2622E-04)**+	1.3246E-02(5.4474E-04)	*Tr-NSGA-II*
	(10,10)	4.4565E-03(3.2910E-04)–	**3.9369E-03(1.7894E-05)**	**2.4904E-03(7.8910E-05)**+	2.6769E-03(4.8851E-05)	**8.4423E-03(2.1685E-04)**+	1.3350E-02(5.7274E-04)	*AE-NSGA-II*
	(10,25)	4.0311E-03(4.1512E-04)≈	**3.7574E-03(5.9892E-06)**	**2.3696E-03(1.8348E-05)**+	2.7010E-03(7.0363E-05)	**8.4499E-03(1.5480E-04)**+	1.3039E-02(5.4375E-04)	*AE-NSGA-II*
	(10,50)	4.0713E-03(3.8327E-04)–	**3.7262E-03(1.0717E-06)**	**2.3350E-03(1.4573E-05)**+	2.6784E-03(6.9388E-05)	**8.2899E-03(1.6327E-04)**+	1.2917E-02(4.8036E-04)	*AE-NSGA-II*
	(1,10)	**1.1259E-01(4.3391E-04)**≈	1.1267E-01(6.2309E-05)	**2.2973E-03(4.6234E-06)**+	1.2888E-01(4.5184E-03)	**1.1543E-01(3.1140E-04)**+	1.1916E-01(5.7331E-04)	*AE-NSGA-II*
	(1,50)	**1.1203E-01(1.7019E-03)**≈	1.1271E-01(3.0409E-05)	**1.1069E-01(5.1236E-04)**+	1.3599E-01(4.4502E-03)	**1.1524E-01(1.6355E-04)**+	1.1867E-01(4.3325E-04)	*AE-NSGA-II*
	(20,10)	4.1375E-03(3.6547E-04)≈	**3.9201E-03(2.1669E-05)**	**2.4801E-03(8.5695E-05)**+	2.7419E-03(3.1451E-04)	**8.3253E-03(1.3495E-04)**+	1.3074E-02(4.4513E-04)	*AE-NSGA-II*
	(20,50)	4.0566E-03(5.5414E-04)≈	**3.7209E-03(1.5564E-06)**	**2.3360E-03(1.2918E-05)**+	2.6215E-03(2.1956E-05)	**8.2572E-03(1.9253E-04)**+	1.2581E-02(4.6943E-04)	*AE-NSGA-II*
dMOP2_{dec}	(10,5)	**3.1166E-01(1.8262E-01)**≈	3.1445E-01(2.1112E-01)	1.2454E+00(3.6801E-01)–	**6.8750E-01(1.1934E-01)**	2.0863E-01(6.8340E-02)≈	**1.8600E-01(4.0582E-02)**	*MDP-MOPSO*
	(10,10)	**7.3751E-02(3.3693E-02)**≈	7.5638E-02(3.0018E-02)	**5.7749E-01(2.6187E-01)**≈	6.3520E-01(8.4463E-02)	**1.8310E-01(7.6834E-02)**≈	1.8558E-01(4.2532E-02)	*AE-MOEA/D*
	(10,25)	**1.9928E-02(6.8989E-03)**≈	2.4293E-02(1.2922E-02)	**2.0311E-01(1.3748E-01)**+	3.8271E-01(2.0092E-02)	2.3438E-01(1.0294E-01)–	**1.5589E-01(3.4968E-02)**	*AE-MOEA/D*
	(10,50)	**1.3149E-02(2.7627E-04)**≈	1.3154E-02(7.1020E-04)	**1.1244E-01(7.5909E-02)**+	1.9893E-01(3.0619E-02)	**1.4902E-01(3.1886E-02)**≈	1.6717E-01(6.7294E-02)	*AE-MOEA/D*
	(1,10)	**4.4064E-01(7.0847E-02)**+	1.3302E+00(3.1744E-01)	**2.1868E+00(3.8111E-01)**+	3.3333E+00(2.1250E-01)	**2.6995E+00(4.6646E-01)**≈	6.4897E+02(1.0438E+03)	*AE-MOEA/D*
	(1,50)	**1.4079E-01(1.4361E-02)**+	1.6797E-01(5.5981E-03)	**6.9309E-01(1.4513E-01)**+	9.4228E-01(6.1623E-02)	**1.2291E+00(2.5750E-01)**+	1.0863E+02(3.4178E+02)	*AE-MOEA/D*
	(20,10)	**4.4419E-02(5.4060E-03)**≈	5.2243E-02(1.4417E-02)	**2.4268E-01(1.0229E-01)**+	6.6107E-01(1.1210E-01)	**1.9235E-01(4.5949E-02)**≈	1.9465E-01(8.0896E-02)	*AE-MOEA/D*
	(20,50)	**1.1693E-02(1.5291E-03)**+	1.2889E-02(1.1268E-03)	**1.2881E-01(8.3161E-02)**+	1.8394E-01(2.3903E-02)	**1.1679E-01(1.9409E-02)**≈	1.1993E-01(1.4730E-02)	*AE-MOEA/D*
dMOP3	(10,5)	**6.4763E-02(3.1273E-02)**≈	9.2821E-02(3.5642E-02)	**4.7352E-01(2.2880E-01)**≈	4.8100E-01(7.8585E-02)	**7.1972E-02(2.5907E-02)**≈	7.4577E-02(3.3035E-02)	*AE-MOEA/D*
	(10,10)	1.9060E-02(9.8874E-03)≈	**1.8050E-02(5.8275E-03)**	**2.1273E-01(1.1741E-01)**+	3.4124E-01(6.8237E-02)	**5.4766E-02(2.2784E-02)**≈	6.3887E-02(2.7020E-02)	*DM-MOEA/D*
	(10,25)	7.0426E-03(2.1219E-03)–	**5.6874E-03(8.5800E-05)**	**8.1259E-02(3.6324E-02)**+	1.2860E-01(1.8447E-02)	**3.4988E-02(1.8904E-02)**≈	3.7250E-02(2.2245E-02)	*DM-MOEA/D*
	(10,50)	5.2812E-03(3.8390E-04)–	**4.5471E-03(1.6376E-05)**	**2.4604E-02(5.5971E-03)**+	3.6965E-02(6.1123E-03)	**2.3298E-02(2.2573E-02)**≈	2.4646E-02(1.3749E-02)	*DM-MOEA/D*
	(1,10)	**1.8284E+01(9.4883E-02)**+	1.9355E+01(2.6445E-01)	**1.9744E-02(8.9195E-05)**+	6.4970E+00(2.5010E+00)	**1.8533E+01(2.7276E-01)**+	2.0295E+01(1.4095E+00)	*AE-NSGA-II*
	(1,50)	1.8011E+01(1.2485E-03)–	**1.8009E+01(1.1842E-05)**	**2.1362E+00(1.8947E+00)**≈	3.2541E+00(1.8786E+00)	**1.8374E+01(2.5601E-01)**≈	1.8676E+01(4.7671E-01)	*AE-NSGA-II*
	(20,10)	**1.0281E-02(4.7358E-03)**≈	1.3246E-02(4.3492E-03)	**1.3251E-01(2.2812E-02)**+	3.6704E-01(9.9133E-02)	1.7626E-01(5.3856E-02)–	**4.3583E-02(2.3002E-02)**	*AE-MOEA/D*
	(20,50)	5.6295E-03(7.8723E-04)–	**4.2367E-03(1.2331E-05)**	**2.5854E-02(9.5528E-03)**+	3.3256E-02(5.0602E-03)	3.9015E-02(1.9932E-02)–	**1.4563E-02(7.6254E-04)**	*DM-MOEA/D*
HE2	(10,5)	**1.8001E-01(3.4648E-02)**+	2.4881E-01(4.2244E-02)	3.2679E-01(1.7667E-01)≈	**2.8894E-01(2.7657E-02)**	**1.3603E-01(4.8002E-03)**≈	1.3617E-01(5.8298E-03)	*AE-MOPSO*
	(10,10)	**1.0039E-01(1.3787E-02)**+	1.4771E-01(2.8702E-02)	**1.8881E-01(8.1097E-02)**+	2.6145E-01(3.4452E-02)	**1.3141E-01(2.1744E-03)**+	1.3393E-01(4.1362E-03)	*AE-MOPSO*
	(10,25)	**6.1084E-02(2.1228E-03)**+	7.6769E-02(5.5121E-03)	**1.1941E-01(3.5704E-02)**+	2.1373E-01(2.4655E-02)	1.3306E-01(4.4461E-03)≈	**1.3117E-01(1.4588E-03)**	*AE-MOEA/D*
	(10,50)	**5.7133E-02(1.7812E-03)**≈	5.8404E-02(1.3637E-03)	**9.4910E-02(1.2410E-02)**+	1.8791E-01(1.0972E-02)	**1.3003E-01(9.0911E-04)**≈	1.3067E-01(2.4586E-03)	*AE-MOEA/D*
	(1,10)	**1.0565E-01(1.0632E-02)**+	1.6584E-01(2.9236E-02)	**7.7647E-02(8.3040E-04)**+	1.0247E-01(3.5501E-02)	1.8751E-01(2.7350E-03)≈	**1.8603E-01(2.0026E-03)**	*AE-NSGA-II*
	(1,50)	5.9769E-02(2.0424E-03)–	**5.7806E-02(5.9393E-04)**	1.1990E-01(2.0275E-02)–	**7.5312E-02(6.8011E-03)**	**1.8404E-01(1.0233E-03)**≈	1.8527E-01(4.0456E-03)	*DM-MOEA/D*
	(20,10)	**9.2434E-02(1.7318E-02)**+	1.4587E-01(1.6897E-02)	**7.3844E-02(7.0653E-03)**+	2.5869E-01(2.6311E-02)	**1.3323E-01(2.2635E-03)**≈	1.3375E-01(2.6741E-03)	*AE-NSGA-II*
	(20,50)	**5.6871E-02(6.6792E-04)**≈	5.7576E-02(5.5528E-04)	**9.4006E-02(3.9025E-02)**+	1.8431E-01(1.2896E-02)	**1.3006E-01(1.0896E-03)**≈	1.3191E-01(1.6278E-03)	*AE-MOEA/D*

nonlinear transfer method (e.g., *Tr-NSGA-II*) is required for prediction.

Next, we summarized the results of the Wilcoxon rank-sum test with the obtained MIGD values of all the compared algorithms on ten DMOPs under eight dynamic configurations (i.e., totally, 80 DMOP instances) in Table V. In the table, each tuple $w/t/l$ denotes the algorithm at the corresponding row wins on w configurations, ties on t configurations, and loses on l configurations, when compared to the algorithms at the corresponding column, respectively. As can be observed, with the proposed autoencoding approach, the multiobjective solver *AE-MOEA/D*, *AE-NSGA-II*, and *AE-MOPSO* achieved significantly superior MIGD values over the corresponding counterpart, on 35, 58, and 42 DMOP instances, respectively.

Last but not least, to investigate the robustness of the proposed approach for handling different dynamic change scenarios, Table VI presents the DMIGD values of all the compared algorithms on all the ten DMOPs. In the table, superior performance is highlighted in bold. It can be observed in the table, using *NSGA-II* and *MOPSO* as the optimization solver, the proposed method performed more robust than the corresponding baseline algorithms, and obtained lower DMIGD values on nine and eight DMOPs, respectively. Furthermore, in the case of considering *MOEA/D* as the optimizer, compared to the more recently proposed DMOEA, that is, *DM-MOEA/D*, the proposed method obtained superior robustness on six out of totally ten DMOPs.

In summary, in terms of solution quality, as *AE-MOEA/D*, *AE-NSGA-II*, and *AE-MOPSO* share the common multiobjective optimizer with the compared counterparts, respectively, and only differ in the prediction method employed for tracking the moving POS, the observed results on MIGD and DMIGD values confirmed the effectiveness and generality of the proposed autoencoding in predicting solutions toward the true POS individuals for tackling DMOPs.

2) Convergence Curve: To access how the proposed approach response when dynamic occurs, here we further plot the convergence curves of the averaged IGD values (over 20 runs) obtained by *AE-MOEA/D*, *DM-MOEA/D*, *AE-NSGA-II*, *Tr-NSGA-II*, *AE-MOPSO*, and *MDP-MOPSO*. In particular, Figs. 3 and 4 present the convergence curves obtained by all the compared algorithms on both the bi-objective and triobjective DMOPs, with $n_t = 10$ and $\tau_t = 10$, respectively. In the figures, the y-axis denotes the averaged IGD values, while the x-axis gives the index of dynamic changes.

TABLE IV
MEAN AND STANDARD DEVIATIONS OF MIGD VALUES OBTAINED BY THE COMPARED ALGORITHMS ON DMOPS POSSESSING THREE OBJECTIVES (SUPERIOR PERFORMANCES ARE HIGHLIGHTED IN BOLD. "+," "−," AND "≈" DENOTE THE PROPOSED AUTOENCODING APPROACH IS STATISTICALLY SIGNIFICANT BETTER, WORSE, AND SIMILAR TO THE COMPARED DYNAMIC MULTIOBJECTIVE METHOD THAT SHARES THE SAME MULTIOBJECTIVE OPTIMIZER, RESPECTIVELY.)

Problems	(n_t,τ_t)	*AE-MOEA/D*	*DM-MOEA/D*	*AE-NSGA-II*	*Tr-NSGA-II*	*AE-MOPSO*	*MDP-MOPSO*	Winner
FDA4	(10,5)	**5.5438E-02(3.0560E-03)**+	6.3129E-02(9.9180E-04)	**2.4727E-01(3.0581E-02)**+	3.2888E-01(1.8109E-02)	**1.0960E-01(2.9296E-03)**+	1.2138E-01(2.6190E-03)	*AE-MOEA/D*
	(10,10)	**5.0357E-02(2.2828E-03)**+	5.4324E-02(5.9963E-04)	**1.9283E-01(1.7996E-02)**+	2.4538E-01(1.9146E-02)	**1.0612E-01(3.9169E-03)**+	1.1799E-01(3.2824E-03)	*AE-MOEA/D*
	(10,25)	**4.7538E-02(2.0324E-03)**+	4.9566E-02(4.5645E-04)	**1.4004E-01(1.1915E-02)**≈	1.4227E-01(7.3432E-03)	**1.0205E-01(2.2251E-03)**+	1.1277E-01(2.2588E-03)	*AE-MOEA/D*
	(10,50)	**4.7358E-02(3.2464E-03)**≈	4.8327E-02(3.1367E-04)	1.1029E-01(3.1124E-03)≈	**1.0812E-01(4.9355E-03)**	**9.5692E-02(2.3679E-03)**+	1.0657E-01(1.8528E-03)	*AE-MOEA/D*
	(1,10)	6.8612E-02(3.4857E-03)–	**5.8265E-02(8.2474E-04)**	**1.2334E-01(1.4628E-17)**+	2.4504E-01(2.1166E-02)	**7.0106E-02(4.3560E-04)**+	9.5487E-02(3.0300E-02)	*MOEA/D-DM*
	(1,50)	**4.7387E-02(2.2863E-03)**+	4.9175E-02(2.0166E-04)	3.1008E-01(3.7931E-02)–	**1.0532E-01(5.0452E-03)**	**6.8066E-02(3.7018E-04)**+	7.6281E-02(1.0914E-03)	*AE-MOEA/D*
	(20,10)	**4.8917E-02(2.1558E-03)**+	5.1826E-02(5.2841E-04)	**1.7500E-01(1.3965E-02)**+	2.5283E-01(1.3100E-02)	**9.0799E-02(8.0714E-04)**+	1.2072E-01(2.6172E-03)	*AE-MOEA/D*
	(20,50)	**4.6741E-02(1.6463E-03)**+	4.8537E-02(2.3030E-04)	**1.0794E-01(4.4749E-03)**≈	1.1209E-01(5.9866E-03)	**8.8958E-02(6.2665E-04)**+	1.0623E-01(2.3446E-03)	*AE-MOEA/D*
FDA5	(10,5)	**8.9458E-02(5.9542E-03)**+	1.0714E-01(3.5552E-03)	**3.1180E-01(4.3605E-02)**+	4.1942E-01(4.2668E-02)	**2.5645E-01(2.3675E-02)**≈	2.7816E-01(3.6711E-02)	*AE-MOEA/D*
	(10,10)	**7.7140E-02(5.4547E-03)**+	8.7306E-02(2.1791E-03)	**2.3491E-01(2.6284E-02)**+	3.2393E-01(4.4685E-02)	**2.1992E-01(1.7271E-02)**+	2.6352E-01(2.8907E-02)	*AE-MOEA/D*
	(10,25)	**7.3470E-02(6.6119E-03)**≈	7.7220E-02(1.3269E-03)	**1.6859E-01(1.1347E-02)**+	2.9216E-01(6.0297E-02)	**2.1156E-01(1.4645E-02)**+	2.4588E-01(2.6074E-02)	*AE-MOEA/D*
	(10,50)	**7.3504E-02(4.6031E-03)**≈	7.4079E-02(1.1395E-03)	**1.2350E-01(6.2110E-03)**+	2.5250E-01(7.1669E-02)	**1.8160E-01(5.8577E-03)**+	2.3337E-01(2.3291E-02)	*AE-MOEA/D*
	(1,10)	2.4923E-01(3.9242E-02)–	**9.2362E-02(7.6257E-03)**	**1.2244E-01(7.4707E-04)**+	4.9366E-01(8.2874E-02)	1.9056E+00(2.7752E-01)–	**2.5798E-01(5.1486E-02)**	*MOEA/D-DM*
	(1,50)	8.9263E-02(4.5583E-02)≈	**6.8445E-02(1.1800E-03)**	**3.7231E-01(8.3528E-02)**≈	3.7758E-01(7.9104E-02)	1.0542E+00(3.5881E-01)–	**1.8220E-01(2.1256E-02)**	*MOEA/D-DM*
	(20,10)	**7.5513E-02(8.1556E-03)**+	8.3909E-02(2.0823E-03)	**1.1449E-01(3.9072E-03)**+	3.0854E-01(3.5653E-02)	**1.6733E-01(1.2775E-02)**+	2.7371E-01(4.4083E-02)	*AE-MOEA/D*
	(20,50)	**7.2381E-02(5.1467E-03)**≈	7.2771E-02(8.9307E-04)	**1.1686E-01(7.0836E-03)**+	2.0702E-01(4.3496E-02)	**1.5461E-01(5.8614E-03)**+	2.1740E-01(1.1917E-02)	*AE-MOEA/D*
FDA5$_{iso}$	(10,5)	**7.6627E-02(3.2182E-03)**+	9.7371E-02(3.9216E-03)	**1.2762E-01(1.2412E-02)**+	7.3921E-01(1.4618E-01)	**1.3824E-01(4.8103E-03)**+	1.8216E-01(1.7561E-02)	*AE-MOEA/D*
	(10,10)	**7.5252E-02(5.5794E-03)**+	8.1802E-02(1.7425E-03)	**1.0574E-01(5.8706E-03)**+	5.5418E-01(8.9131E-02)	**1.3618E-01(5.6053E-03)**+	1.7795E-01(1.1577E-02)	*AE-MOEA/D*
	(10,25)	**7.4516E-02(5.8249E-03)**≈	7.5444E-02(7.4394E-04)	**9.1363E-02(2.3724E-03)**+	2.7436E-01(3.2952E-02)	**1.3808E-01(4.2341E-03)**+	1.7727E-01(6.4125E-03)	*AE-MOEA/D*
	(10,50)	7.3793E-02(6.0000E-03)≈	**7.2547E-02(9.6478E-04)**	**8.5949E-02(2.2347E-03)**+	1.7791E-01(3.3955E-02)	**1.3032E-01(3.2459E-03)**+	1.7912E-01(8.5722E-03)	*MOEA/D-DM*
	(1,10)	9.7966E-02(5.0195E-02)≈	**8.7447E-02(4.5157E-03)**	**7.0582E-02(1.0334E-04)**+	2.6774E-01(3.0034E-02)	2.0975E-01(1.9876E-02)≈	**1.9894E-01(2.3949E-02)**	*AE-NSGA-II*
	(1,50)	1.7245E-01(1.4198E-01)≈	**6.7882E-02(8.4200E-04)**	1.7194E-01(1.3485E-02)≈	**1.5770E-01(1.4304E-02)**	**1.5870E-01(1.4586E-02)**≈	1.7688E-01(2.7535E-02)	*MOEA/D-DM*
	(20,10)	**7.3582E-02(5.6720E-03)**+	7.9803E-02(1.7561E-03)	**7.1102E-02(4.5551E-04)**+	2.8188E-01(6.0804E-02)	**1.3247E-01(3.4279E-03)**+	1.7317E-01(8.9267E-03)	*AE-MOEA/D*
	(20,50)	**7.2160E-02(6.1719E-03)**≈	7.2233E-02(5.9950E-04)	**8.5390E-02(1.1845E-03)**+	1.5815E-01(1.6821E-02)	**1.2888E-01(2.3173E-03)**+	1.7408E-01(9.4669E-03)	*AE-MOEA/D*
FDA5$_{dec}$	(10,5)	**9.6893E-02(5.4887E-03)**+	1.2212E-01(1.0843E-02)	**3.2694E-01(3.4657E-02)**+	5.4112E-01(8.9479E-02)	**2.1124E-01(2.4973E-02)**+	3.5066E-01(9.7338E-02)	*AE-MOEA/D*
	(10,10)	**8.5690E-02(5.8812E-03)**+	9.7149E-02(2.5349E-03)	**2.3819E-01(3.7753E-02)**+	4.4579E-01(4.1175E-02)	**1.9913E-01(1.6740E-02)**+	3.6400E-01(9.0268E-02)	*AE-MOEA/D*
	(10,25)	**8.2407E-02(5.6776E-03)**≈	8.4230E-02(1.2445E-03)	**2.0266E-01(3.1422E-02)**+	3.8836E-01(3.4155E-02)	**1.7874E-01(1.1634E-02)**+	3.1152E-01(9.8820E-02)	*AE-MOEA/D*
	(10,50)	**8.0137E-02(5.7864E-03)**≈	8.0503E-02(8.3579E-04)	**1.4860E-01(1.3162E-02)**+	3.2701E-01(3.9871E-02)	**1.6586E-01(1.2117E-02)**+	2.9073E-01(1.0775E-01)	*AE-MOEA/D*
	(1,10)	6.5977E-01(2.2189E-01)–	**1.7700E-01(1.6254E-02)**	**2.0115E-01(8.6890E-03)**+	1.1026E+00(1.6436E-01)	1.2742E+00(1.3904E+00)≈	**9.0817E-01(1.2742E-01)**	*MOEA/D-DM*
	(1,50)	7.2536E-01(1.3411E-01)–	**7.9867E-02(1.0802E-03)**	**3.5635E-01(7.0103E-02)**+	7.9094E-01(1.3564E-01)	1.0283E+00(3.1784E-01)≈	**7.8758E-01(1.4670E-01)**	*MOEA/D-DM*
	(20,10)	**8.3755E-02(2.0006E-03)**+	9.4325E-02(3.1998E-03)	**2.2087E-01(1.9586E-02)**+	4.1734E-01(3.0286E-02)	**1.8319E-01(1.2937E-02)**+	3.2064E-01(1.2224E-01)	*AE-MOEA/D*
	(20,50)	**7.6494E-02(4.0386E-03)**+	8.0044E-02(8.0870E-04)	**1.6182E-01(2.4241E-02)**+	2.6443E-01(3.9545E-02)	**1.5539E-01(1.0799E-02)**+	2.4403E-01(9.2257E-02)	*AE-MOEA/D*

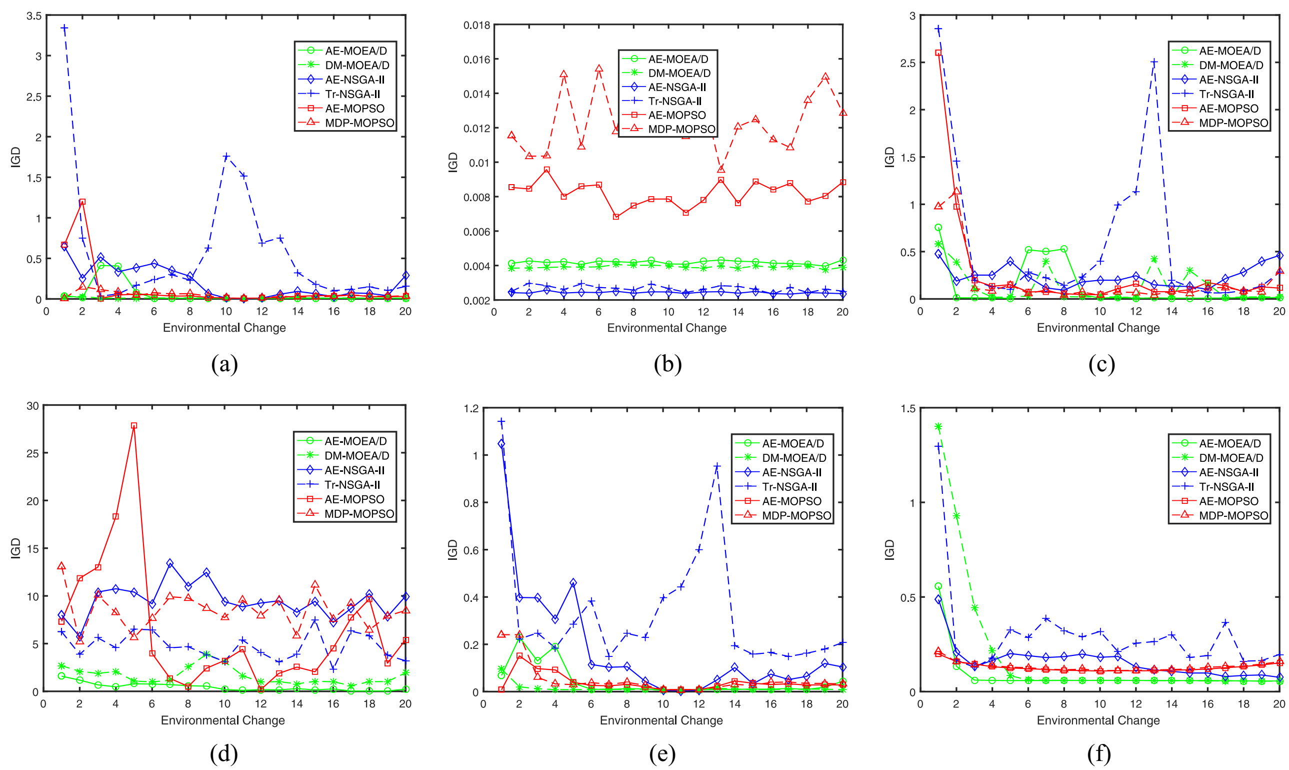

Fig. 3. Convergence curves of the averaged IGD (over 20 runs) obtained by *AE-MOEA/D*, *DM-MOEA/D*, *AE-NSGA-II*, *Tr-NSGA-II*, *AE-MOPSO*, and *MDP-MOPSO* on the biobjective DMOPs, with $n_t = 10$ and $\tau_t = 10$ (y-axis: the IGD value and x-axis: the index of dynamic change). (a) dMOP2. (b) dMOP2$_{iso}$. (c) dMOP2$_{\text{dec}}$. (d) DIMP2. (e) dMOP3. (f) HE2.

It can be observed in Figs. 3 and 4 that the dynamic changes of the problem always lead to an increase in the IGD values if the tracking of dynamic is not accurately given by the prediction methods. Generally, on all the studied DMOPs, the proposed autoencoding-based prediction achieved superior performance in terms of tracking dynamic

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

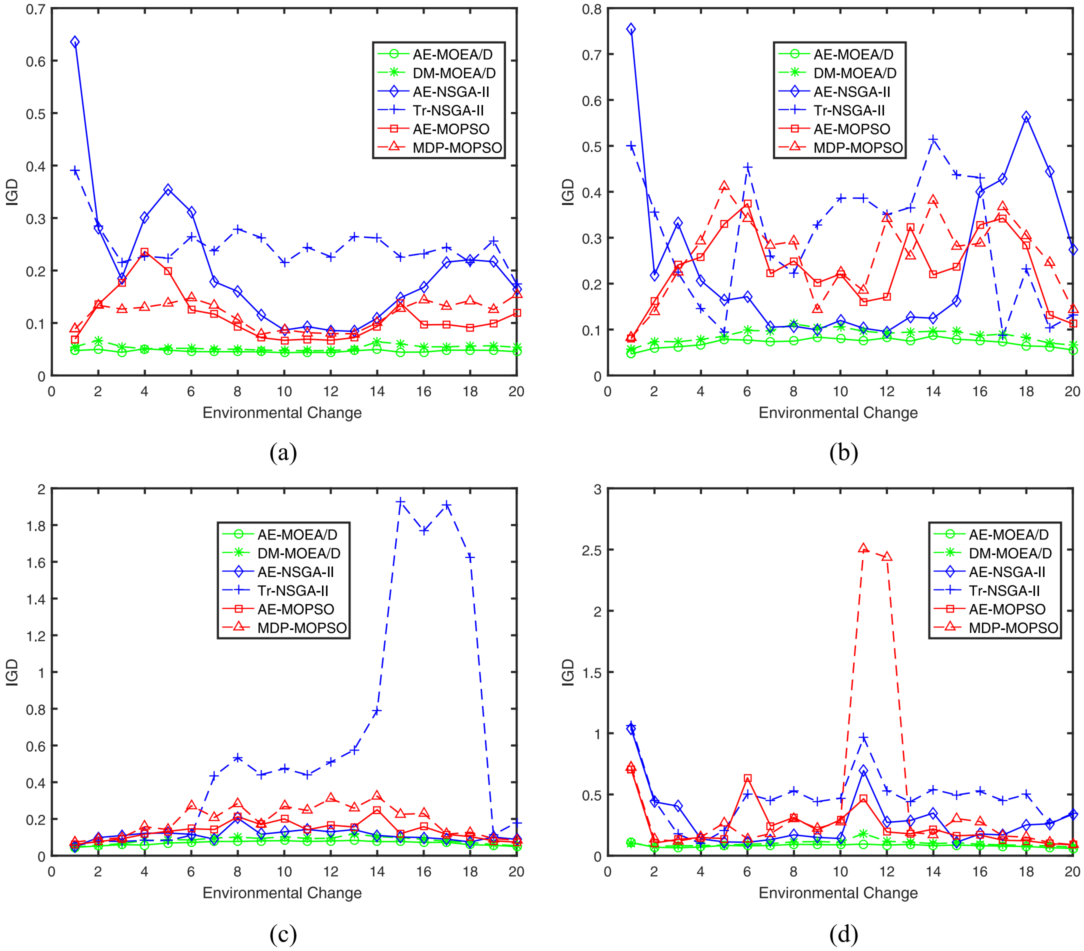

Fig. 4. Convergence curves of the averaged IGD (over 20 runs) obtained by *AE-MOEA/D*, *DM-MOEA/D*, *AE-NSGA-II*, *Tr-NSGA-II*, *AE-MOPSO*, and *MDP-MOPSO* on the triobjective DMOPs, with $n_t = 10$ and $\tau_t = 10$ (y-axis: the IGD value and x-axis: the index of dynamic change). (a) FDA4. (b) FDA5. (c) FDA5$_{iso}$. (d) FDA5$_{\text{dec}}$.

TABLE V
SUMMARIZED MIGD RESULTS BETWEEN THE PROPOSED AUTOENCODING APPROACH AND THE COMPARED DMOEAS ON 80 CONFIGURATIONS OF THE DMOPS (EACH TUPLE $w/t/l$ DENOTES THE ALGORITHM AT THE CORRESPONDING ROW WINS ON w CONFIGURATIONS, TIES ON t CONFIGURATIONS, AND LOSES ON l CONFIGURATIONS, WHEN COMPARED TO THE ALGORITHMS AT THE CORRESPONDING COLUMN, RESPECTIVELY.)

	DM-MOEA/D	*Tr-NSGA-II*	*MDP-MOPSO*
AE-MOEA/D	35/32/13		
AE-NSGA-II		58/10/12	
AE-MOPSO			43/29/8

changes, in contrast to the compared prediction approaches. In particular, using *MOPSO* as the optimizer, *AE-MOPSO* and *MDP-MOPSO* performs competitively on DMOPs, such as dMOP2$_{\text{dec}}$, dMOP3, and HE2. However, on DMOPs, such as dMOP2$_{iso}$, DMIP2, FDA5$_{\text{dec}}$, etc., *AE-MOPSO* outperforms *MDP-MOPSO* with lower IGD values obtained when dynamic occurs. Furthermore, using *NSGA-II* as the optimization solver, similar observation can be observed between *AE-NSGA-II* and *Tr-NSGA-II*. On DMOPs, such as DMOP3, FDA4, and FDA5, significant superior predictions (i.e., lower IGD values) have been obtained by *AE-NSGA-II* with the proposed prediction approach over *Tr-NSGA-II*, from dynamic change 8 to 14. Nevertheless, similar to the observation in Table III, due to the different change rates of decision variables in DIMP2, *Tr-NSGA-II* obtained lower IGD values when dynamic occurs on this problem. Finally, in the comparison of using *MOEA/D* as the evolutionary solver, *AE-MOEA/D* obtained superior or competitive performance against *DM-MOEA/D* on all the DMOPs. Particularly, on the complex DIMP2, *AE-MOEA/D* achieved consistently better predictions of POS (i.e., with lower IGD value) over *DM-MOEA/D* along the evolutionary search process.

3) Running Time: Next, Figs. 5 and 6 present the running time (in seconds) of all the compared algorithms on two representative DMOPs, that is, dMOP2 and HE2. As can be observed in the figures, on dMOP2, which has ten number of variables (see Table I), the algorithms *AE-MOEA/D* and *AE-MOPSO* used similar running time in contrast to the baseline algorithm, that is, *DM-MOEA/D* and *MDP-MOPSO*, respectively. For the case of using *NSGA-II* as the basic optimizer, the proposed prediction method (i.e., *AE-NSGA-II*) only took 38 s while the compared *Tr-NSGA-II* used over 1240 s, to

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VI
DMIGD VALUES OF THE COMPARED ALGORITHMS (SUPERIOR PERFORMANCE IS HIGHLIGHTED IN BOLD.)

DMIGD	*AE-MOEA/D*	*DM-MOEA/D*	*AE-NSGA-II*	*Tr-NSGA-II*	*AE-MOPSO*	*MDP-MOPSO*	Winner
FDA4	**5.1543E-02**	5.2894E-02	**1.7585E-01**	1.9249E-01	**9.1425E-02**	1.0718E-01	*AE-MOEA/D*
FDA5	9.9995E-02	**8.2905E-02**	**1.9561E-01**	3.3435E-01	5.1891E-01	**2.4403E-01**	*DM-MOEA/D*
FDA5_{iso}	8.9543E-02	**7.9316E-02**	**1.0121E-01**	3.2639E-01	**1.4658E-01**	1.7995E-01	*DM-MOEA/D*
FDA5_{dec}	2.3631E-01	**1.0191E-01**	**2.3207E-01**	5.3470E-01	1.4448E+01	**4.4717E-01**	*DM-MOEA/D*
DIMP2	**4.6338E-01**	1.0545E+00	6.3871E+00	**3.7953E+00**	**6.7014E+00**	7.0764E+00	*AE-MOEA/D*
dMOP2	**4.5703E+00**	4.5923E+00	**3.6164E-01**	1.5870E+00	**4.8118E+00**	5.1655E+00	*AE-NSGA-II*
dMOP2_{iso}	3.1205E-02	**3.1101E-02**	**1.6007E-02**	3.5117E-02	**3.5108E-02**	3.9505E-02	*AE-NSGA-II*
dMOP2_{dec}	**1.3200E-01**	2.4886E-01	**6.7372E-01**	8.7812E-01	**6.2661E-01**	9.4826E+01	*AE-MOEA/D*
dMOP3	**4.5509E+00**	4.6879E+00	**3.8831E-01**	1.3924E+00	**4.6634E+00**	4.9036E+00	*AE-NSGA-II*
HE2	**8.9168E-02**	1.1985E-01	**1.3691E-01**	1.9660E-01	**1.4603E-01**	1.4611E-01	*AE-MOEA/D*

TABLE VII
MEAN AND STANDARD DEVIATIONS OF MIGD VALUES OBTAINED BY THE COMPARED ALGORITHM ON DMOPS WITH $n_t = 1$ AND $\tau_t = 10$ (SUPERIOR PERFORMANCES ARE HIGHLIGHTED IN BOLD. "+," "−," AND "≈" DENOTE THE PROPOSED AUTOENCODING APPROACH IS STATISTICALLY SIGNIFICANT BETTER, WORSE, AND SIMILAR TO THE COMPARED DYNAMIC MULTIOBJECTIVE METHOD THAT SHARES THE SAME MULTIOBJECTIVE OPTIMIZER, RESPECTIVELY.)

Problems	(n_t,τ_t)	*AE-MOEA/D*	*DM-MOEA/D*	*AE-NSGA-II*	*Tr-NSGA-II*	*AE-MOPSO*	*MDP-MOPSO*	Winner
DF2	(1,10)	**1.8588E-01 (5.3996E-02)**+	4.0425E-01 (3.5060E-02)	**3.8926E-02 (7.6778E-03)**+	1.9885E-01 (3.3796E-02)	2.3671E-01 (1.6270E-01)≈	**2.1988E-01 (6.7183E-02)**	*AE-NSGA-II*
DF3	(1,10)	**3.2856E-01 (3.8766E-02)**+	4.0425E-01 (4.4426E-02)	**4.3977E-02 (1.2722E-02)**+	1.7948E-01 (1.8501E-02)	**4.7168E-01 (4.7970E-02)**≈	5.5480E-01 (9.7972E-02)	*AE-NSGA-II*
DF4	(1,10)	**3.1926E-02 (6.5200E-03)**≈	3.4391E-02 (3.0115E-03)	**4.4130E-02 (1.5857E-02)**+	8.8512E-01 (4.3904E-02)	**1.1727E+00 (1.0660E-01)**+	1.3074E+00 (1.3383E-01)	*AE-MOEA/D*
DF5	(1,10)	**1.2039E-02 (8.4812E-04)**+	2.6186E-02 (7.6505E-03)	**2.9331E-02 (1.8734E-02)**+	6.4382E-01 (2.0768E-01)	1.7240E-01 (2.7714E-02)≈	**1.6015E-01 (3.2772E-02)**	*AE-MOEA/D*
DF6	(1,10)	**5.2027E-01 (6.1974E-02)**≈	5.2167E-01 (6.1226E-02)	**8.1077E-01 (1.4760E-01)**+	1.0004E+00 (2.3585E-01)	**1.1148E+00 (3.6485E-01)**≈	1.1313E+00 (5.3687E-01)	*AE-MOEA/D*
DF7	(1,10)	1.5418E+00 (2.5378E-01)–	**1.1377E+00 (1.5832E-01)**	**1.6442E-02 (3.2653E-03)**+	4.0585E-01 (7.6556E-02)	**4.1082E-01 (3.4018E-02)**≈	4.4562E-01 (5.3936E-02)	*AE-NSGA-II*
DF8	(1,10)	1.3256E-01 (6.8026E-02)–	**5.4880E-02 (8.2769E-04)**	**9.4800E-03 (1.3647E-03)**+	1.1721E-01 (1.2036E-02)	**1.1281E-01 (6.1187E-03)**+	2.1172E-01 (1.0065E-02)	*AE-NSGA-II*
DF9	(1,10)	**4.1089E-01 (5.4739E-02)**+	5.7850E-01 (7.8219E-02)	**8.3976E-02 (8.9266E-03)**+	1.4095E-01 (2.0269E-02)	**4.7054E-01 (3.5455E-02)**≈	4.7206E-01 (2.5067E-02)	*AE-NSGA-II*
DF10	(1,10)	**9.3651E-02 (1.1005E-02)**+	1.7813E-01 (1.8818E-02)	**1.0608E-01 (5.2002E-03)**+	4.2521E-01 (8.4440E-03)	**2.0123E-01 (7.2611E-03)**+	2.2176E-01 (4.2698E-03)	*AE-MOEA/D*
DF11	(1,10)	**5.4219E-01 (9.4470E-04)**≈	5.4230E-01 (9.4781E-04)	**1.2461E-01 (1.1065E-02)**+	3.1858E+01 (1.0923E-01)	**5.7843E-01 (4.9455E-03)**+	6.0116E-01 (6.3307E-03)	*AE-NSGA-II*
DF12	(1,10)	**2.1326E-01 (6.6561E-02)**+	2.2116E-01 (5.7936E-03)	**9.7510E-02 (3.9438E-03)**+	3.4304E-01 (2.1138E-02)	**3.4486E-01 (5.0788E-02)**+	5.5100E-01 (4.1347E-02)	*AE-NSGA-II*
DF13	(1,10)	2.1362E-01 (1.2785E-02)–	**1.9817E-01 (4.1290E-03)**	**2.8531E-01 (1.8562E-02)**+	1.1136E+00 (2.3781E-01)	5.9460E-01 (6.1024E-02)≈	**5.5553E-01 (4.4171E-02)**	*DM-MOEA/D*
DF14	(1,10)	**3.3927E-02 (7.5018E-03)**≈	3.4242E-02 (8.1230E-04)	**8.3455E-02 (2.5336E-02)**+	2.5226E-01 (1.2714E-02)	**2.5401E-01 (5.8523E-02)**≈	2.8252E-01 (1.0194E-01)	*AE-MOEA/D*
w / t / l		6 / 4 / 3		13 / 0 / 0		5 / 8 / 0		

$w/t/l$ indicates that the proposed autoencoding approach wins on w configurations, ties on t configurations, and loses on l configurations, when compared to the algorithm which possesses the same evolutionary solver, respectively.

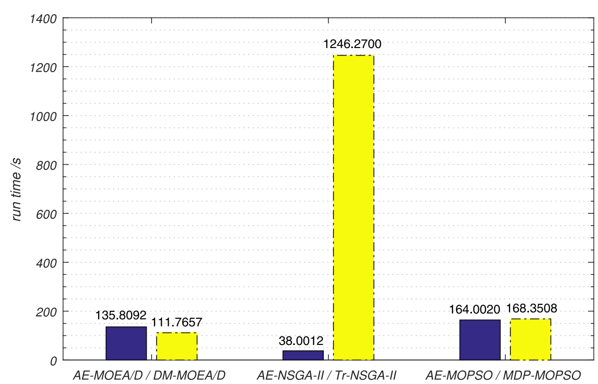

Fig. 5. Running time (in seconds) of all the compared algorithms on DMOP dMOP2.

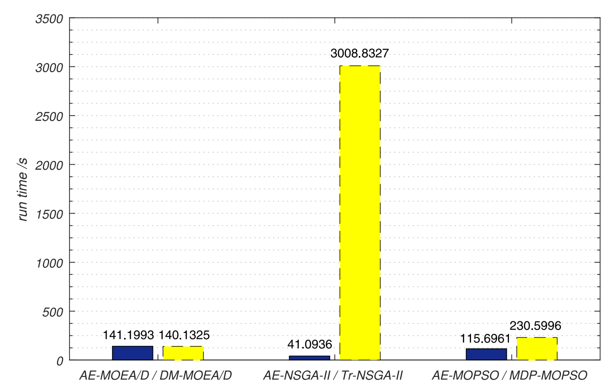

Fig. 6. Running time (in seconds) of all the compared algorithms on DMOP HE2.

solve dMOP2. Furthermore, on the DMOP HE2, which is a 30-D DMOP, the comparisons between the algorithms using *MOEA/D* and *NSGA-II* as the basic evolutionary solver, are consistent with that observed on DMOP dMOP2. However, when compared to *MDP-MOPSO*, the proposed prediction method, that is, *AE-MOPSO*, is more efficient when the number of decision variables increases.

4) Predicted Initial Solutions: Furthermore, to provide deeper insights into the superior performance obtained by

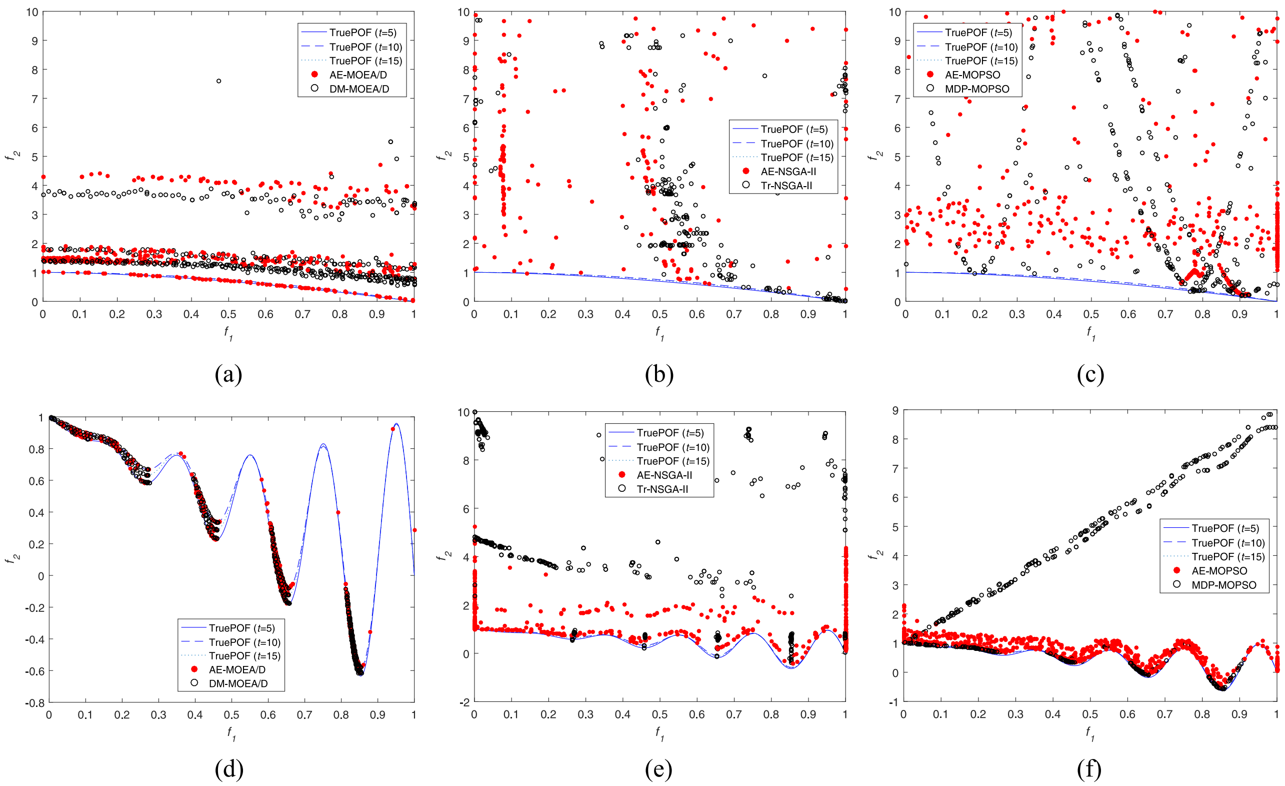

Fig. 7. Predicted solutions obtained by the proposed *AE-MOEA/D*, *AE-NSGA-II*, and *AE-MOPSO* against the compared baseline algorithm on DMOP $\text{dMOP2}_{\text{dec}}$ and HE2, at different experimental change t. (a) $\text{dMOP2}_{\text{dec}}$. (b) $\text{dMOP2}_{\text{dec}}$. (c) $\text{dMOP2}_{\text{dec}}$. (d) HE2. (e) HE2. (f) HE2.

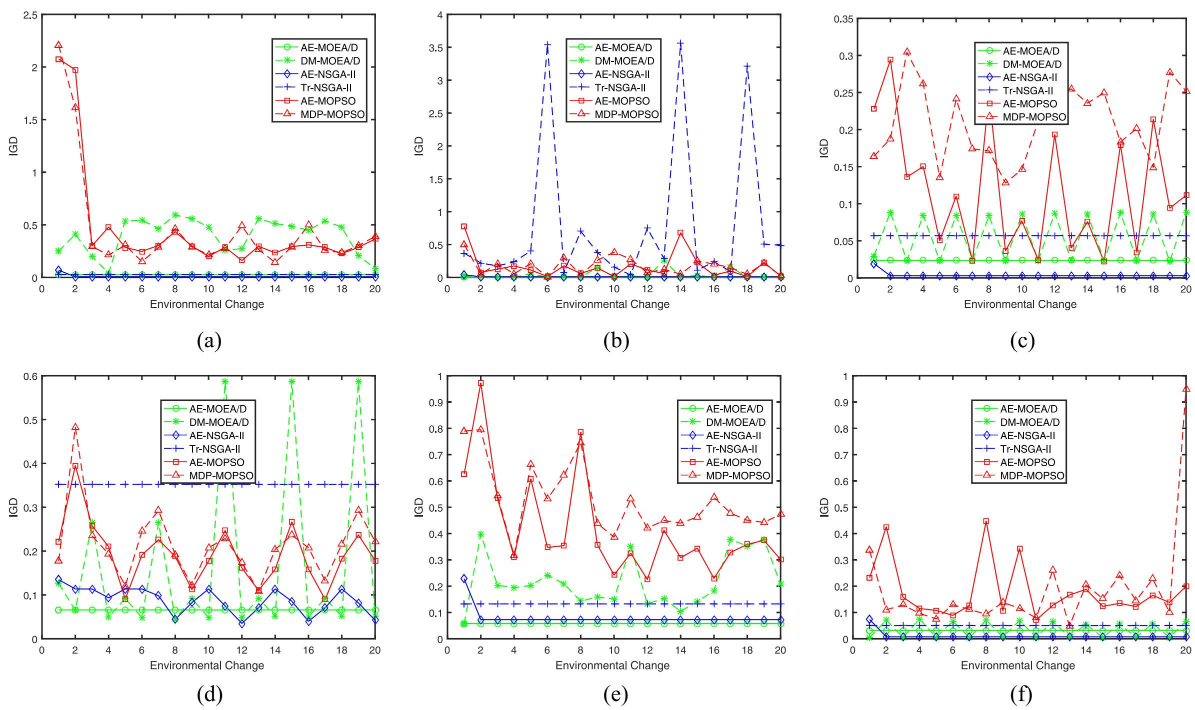

Fig. 8. Convergence curves of the averaged IGD (over 20 runs) obtained by *AE-MOEA/D, DM-MOEA/D, AE-NSGA-II, Tr-NSGA-II, AE-MOPSO,* and *MDP-MOPSO* on both the biobjective and triobjective DMOPs, with $n_t = 1$ and $\tau_t = 10$ (y-axis: the IGD value and x-axis: the index of dynamic change). (a) DF3. (b) DF5. (c) DF8. (d) DF10. (e) DF12. (f) DF14.

the proposed autoencoding-based prediction, for tracking the dynamic changes in DMOP, we further plot the predicted solutions of all the compared algorithms when dynamic occurs, on representative DMOPs, in Fig. 7. These predicted solutions explained why the proposed method obtained a superior performance when compared to the other three approaches for solving DMOPs, as discussed in Sections IV-B1 and IV-B2. In particular, as can be observed in the figures, predicted solutions obtained by the proposed method (denoted by red dots in the figures) are more diverse and close to the true POF. For example, in Fig. 7(b), the predicted solutions of *Tr-NSGA-II* are mostly having f_1 values larger than 0.4, while the solutions obtained by the proposed method distributed along the f_1 axis, ranging from 0 to 0.9. In Fig. 7(d), only the proposed method (i.e., *AE-MOEA/D*) contains the predicted solution with f_1 and f_2 values larger than 0.9 and 0.8, respectively. Further, in Fig. 7(f), it can be observed that the predicted solutions in *AE-MOPSO* have a better distribution in contrast to the solutions of *MDP-MOPSO*, with respect to the true POF at different experimental change t.

5) Further Comparison on CEC2018 Benchmarks: Last but not least, we further evaluate the efficacy of the proposed autoencoding-based prediction on the recently developed IEEE CEC2018 DMOP benchmark functions [56]. In particular, the configurations of dynamic changes are set as $n_t = 1$ and $\tau_t = 10$, which indicate a large and frequent dynamic change in the DMOPs. Other parameter and operator configurations of the compared algorithms are kept consistent with the empirical studies conducted above.

Table VII summarizes the MIGD values obtained by all the compared algorithms over 20 independent runs, on the DMOPs from the IEEE CEC 2018 dynamic benchmark problems. As the DMOP "DF1" is the same as "dMOP2" we investigated above [56], here we present the other 13 CEC 2018 benchmarks in Table VII. In the table, superior performances are highlighted in bold, and the Wilcoxon rank-sum test with 95% confidence level is also conducted on the experimental results. Moreover, Fig. 8 illustrates the convergence curves of the averaged IGD values (over 20 runs) obtained by all the compared algorithms on representative DMOPs. In Table VII and Fig. 8, similar to the observations obtained on the IEEE CEC 2015 benchmarks, the proposed autoencoding-based prediction achieved superior performance in terms of both solution quality and tracking dynamic changes, compared to all the baseline algorithms on most of the DMOPs.

V. Conclusion

In this article, to solve DMOP, we have proposed an autoencoding evolutionary search approach for tracking the moving direction of POS while the evolutionary search progresses online. The proposed prediction approach can be easily integrated into the existing static MOEAs. In particular, the proposed prediction method mainly contains two components, one is *prediction via denoising autoencoding*, and the other is *high-quality solution preservation*. For *prediction via denoising autoencoding*, a single-layer denoising autoencoder that holds a closed-form solution has been derived. As the proposed prediction is automatically learned from the nondominated solutions found along the dynamic optimization process, it is able to provide a more accurate prediction of moving POS. Furthermore, the *high-quality solution preservation*, on the other hand, have been proposed to preserve the nondominated solutions found along the optimization process. To evaluate the performance of the proposed approach, comprehensive empirical studies have been conducted on ten commonly used DMOPs under eight different dynamic configurations. The results obtained by comparing against three state-of-the-art DMOEAs, which possess different multiobjective solvers, have confirmed the efficacy of the proposed autoencoding evolutionary search for tackling DMOPs. Particularly, on totally 80 number of DMOP instances, the proposed method is found to obtain superior or competitive MIGD values over the baseline predictions methods in [14], [42], and [41] on 67, 68, and 72 DMOP instances, respectively, using much less or competitive computational cost.

For future work, we would like to continue the deeper analysis of the autoencoder-based prediction toward more accurate tracking of moving POS, for efficient and effective solving of DMOPS. For instance, the learning of mapping from the space of POF could complement the prediction in the space of POS. Furthermore, we would also like to apply the proposed method to solve real-world DMOP applications, such as optimization of robot action, flight trajectory planning, etc.

References

[1] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization," in *Proc. 5th Int. Conf. Genet. Algorithms*, 1993, pp. 416–423.

[2] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering. structural and multidisciplinary optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 6, pp. 369–395, Mar. 2004.

[3] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[4] C. A. Coello Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.

[5] S. Bechikh, R. Datta, and A. Gupta, *Recent Advances in Evolutionary Multi-Objective Optimization*. Cham, Switzerland: Springer, 2017.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[7] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Dept. Elect. Eng., Swiss Federal Inst. Technol., Zürich, Switzerland, TIK Rep. 103, 2001.

[8] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[9] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440–462, Jun. 2017.

[10] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

[11] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via Kalman filter prediction," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2862–2873, Dec. 2016.

[12] S. B. Gee, K. C. Tan, and C. Alippi, "Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4223–4234, Dec. 2017.

[13] R. Chen, K. Li, and X. Yao, "Dynamic multiobjectives optimization with a changing number of objectives," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 157–171, Feb. 2018.

[14] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer learning-based dynamic multiobjective optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 501–514, Aug. 2018.

[15] Q. Zhang, S. Yang, S. Jiang, R. Wang, and X. Li, "Novel prediction strategies for dynamic multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 260–274, Apr. 2020.

[16] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, "A similarity-based cooperative co-evolutionary algorithm for dynamic interval multi-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 142–156, Feb. 2020.

[17] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, "A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, early access, Apr. 24, 2019, doi: 10.1109/TCYB.2019.2909806.

[18] S. Yang and X. Yao, *Evolutionary Computation for Dynamic Optimization Problems*. Heidelberg, Germany: Springer Int., 2013.

[19] Radhia R. Azzouz, S. Bechikh, and L. B. Said, *Dynamic Multi-objective Optimization Using Evolutionary Algorithms: A Survey*. Cham, Switzerland: Springer Int., 2017, pp. 31–70.

[20] K. Deb, N. U. B. Rao, and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2007, pp. 803–817.

[21] J. Grefenstette. "Genetic algorithms for changing environments," in *Proc. Parallel Problem Solv. Nat.*, 1992, pp. 137–144.

[22] H. G. Cobb and J. J. Grefenstette, "Genetic algorithms for tracking changing environments," in *Proc. 5th Int. Conf. Genet. Algorithms*, 1993, pp. 523–530.

[23] C. Li and S. Yang, "A general framework of multipopulation methods with clustering in undetectable dynamic environments," *IEEE Trans. Evol. Comput.*, vol. 16, no. 4, pp. 556–577, Aug. 2012.

[24] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[25] A. Toffolo and E. Benini, "Genetic diversity as an objective in multi-objective evolutionary algorithms," *Evol. Comput.*, vol. 11, no. 2, pp. 151–167, 2003.

[26] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, Jan. 2014.

[27] A. Muruganantham, K. C. Tan, and P. Vadakkepat, *Solving the IEEE CEC 2015 Dynamic Benchmark Problems Using Kalman Filter Based Dynamic Multiobjective Evolutionary Algorithm*. Cham, Switzerland: Springer Int., 2016.

[28] I. Hatzakis and D. Wallace, "Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach," in *Proc. 8th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2006, pp. 1201–1208.

[29] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, "Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 305–319, Apr. 2020.

[30] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.

[31] Y. Wang and B. Li, "Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization," *Memetic Comput.*, vol. 2, no. 1, pp. 3–24, 2010.

[32] D. E. Goldberg and R. E. Smith, "Nonstationary function optimization using genetic algorithm with dominance and diploidy," in *Proc. 2nd Int. Conf. Genet. Algorithms Appl.*, 1987, pp. 59–68.

[33] J. Lewis, E. Hart, and G. Ritchie, "A comparison of dominance mechanisms and simple mutation on non-stationary problems," in *Proc. 5th Int. Conf. Parallel Problem Solv. Nat. (PPSN)*, 1998, pp. 139–148.

[34] N. Mori, H. Kita, and Y. Nishikawa, "Adaptation to a changing environment by means of the feedback thermodynamical genetic algorithm," in *Proc. 5th Int. Conf. Parallel Problem Solv. Nat. (PPSN)*, 1998, pp. 149–158.

[35] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 3, 1999, pp. 1875–1882.

[36] S. Yang, "Non-stationary problem optimization using the primal-dual genetic algorithm," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 3, 2003, pp. 2246–2253.

[37] M. Camara, J. Ortega, and F. J. Toro, "Parallel processing for multi-objective optimization in dynamic environments," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, 2007, pp. 1–8.

[38] F. O. De Françaand F. J. Von Zuben, "A dynamic artificial immune algorithm applied to challenging benchmarking problems," in *Proc. Congr. Evol. Comput.*, 2009, pp. 423–430.

[39] C. Raquel and X. Yao, "Dynamic multi-objective optimization: A survey of the state-of-the-art," in *Evolutionary Computation for Dynamic Optimization Problems*. Heidelberg, Germany: Springer, 2013, pp. 85–106.

[40] J. Zou, Q. Li, S. Yang, H. Bai, and J. Zheng, "A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization," *Appl. Soft Comput.*, vol. 61, pp. 806–818, 2017.

[41] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, "Multidirectional prediction approach for dynamic multiobjective optimization problems," *IEEE Trans. Cybern.*, pp. 1–13, 2018.

[42] L. Cao, L. Xu, E. D. Goodman, and H. Li, "Decomposition-based evolutionary dynamic multiobjective optimization using a difference model," *Appl. Soft Comput.*, vol. 76, pp. 473–490, 2019.

[43] L. Feng, Y. S. Ong, S. Jiang, and A. Gupta, "Autoencoding evolutionary search with learning across heterogeneous problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 760–772, Oct. 2017.

[44] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2006, pp. 153–160.

[45] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[46] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.

[47] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 833–840.

[48] Y. Sun, H. Mao, Q. Guo, and Z. Yi, "Learning a good representation with unsymmetrical auto-encoder," *Neural Comput. Appl.*, vol. 27, no. 5, pp. 1361–1367, 2016.

[49] Y. Sun, H. Mao, Y. Sang, and Z. Yi, "Explicit guiding auto-encoders for learning meaningful representation," *Neural Comput. Appl.*, vol. 28, no. 3, pp. 429–436, 2017.

[50] H. Richter, "Detecting change in dynamic fitness landscapes," in *Proc. IEEE Congr. Evol. Comput.*, 2009, pp. 1613–1620.

[51] M. Helbig and A. Engelbrecht, "Benchmark functions for CEC 2015 special session and competition on dynamic multi-objective optimization," Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, Rep., 2015.

[52] S. Jiang and S. Yang, "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 198–211, Jan. 2017.

[53] S. Jiang, M. Kaiser, S. Yang, S. Kollias, and N. Krasnogor, "A scalable test suite for continuous dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2814–2826, Jun. 2020.

[54] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, Oct. 2004.

[55] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 65–82, Feb. 2017.

[56] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser, and N. Krasnogor, "Benchmark problems for CEC2018 competition on dynamic multiobjective optimisation," Dept. Comput. Sci., Newcastle Univ., Newcastle upon Tyne, U.K., Rep., 2018.

Liang Feng received the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2014.

He is currently a Professor with the College of Computer Science, Chongqing University, Chongqing, China. His research interests include computational and artificial intelligence, memetic computing, big data optimization and learning, as well as transfer learning.

Dr. Feng's research work on evolutionary multitasking won the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award in 2019. He is an Associate Editor of the *IEEE Computational Intelligence Magazine*, *Memetic Computing*, and *Cognitive Computation*. He is also the Founding Chair of the IEEE CIS Intelligent Systems Applications Technical Committee Task Force on "Transfer Learning and Transfer Optimization."

Wei Zhou received the B.E. degree from the Department of Computer Science, Chongqing University, Chongqing, China, in 2016, where he is currently pursuing the Ph.D. degree.

His current research interests include dynamic multiobjective optimization and evolutionary transfer optimization.

Weichen Liu received the B.S. and M.S. degrees from Harbin Institute of Technology, Harbin, China, in 2004 and 2006, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2011.

He is a Nanyang Assistant Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has authored and coauthored more than 120 publications in peer-reviewed journals, conferences, and books. His research interests include embedded and real-time systems, many-core systems, electronic/photonic network-on-chip, machine learning acceleration, photonic computing, and in-memory computing.

Yew-Soon Ong (Fellow, IEEE) received the Ph.D. degree in artificial intelligence in complex design from the Computational Engineering and Design Center, University of Southampton, Southampton, U.K., in 2003.

He is a Professor and the Chair of the School of Computer Science and Engineering, Nanyang Technological University, Singapore, where he is the Director of the Data Science and Artificial Intelligence Research Center and the A*Star SIMTECH-NTU Joint Laboratory on Complex Systems, and the Principal Investigator of the Data Analytics and Complex System Programme, Rolls-Royce@NTU Corporate Laboratory.

Prof. Ong is the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE and an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON NEURAL NETWORK AND LEARNING SYSTEMS, and the IEEE TRANSACTIONS ON CYBERNETICS.

Kay Chen Tan (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published over 200 refereed articles and six books.

Prof. Tan is the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He was the Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2010 to 2013, and currently serves as the Editorial Board Member of over ten journals. He was an Elected Member of the IEEE CIS AdCom from 2017 to 2019.