MOEA/D With Spatial–Temporal Topological Tensor Prediction for Evolutionary Dynamic Multiobjective Optimization

Xianpeng Wang, *Senior Member, IEEE*, Yumeng Zhao, Lixin Tang, *Fellow, IEEE*, and Xin Yao, *Fellow, IEEE*

***Abstract*—When solving dynamic multiobjective optimization problems, most evolutionary algorithms (EAs) attempt to predict the initial population in a new environment by mining the relationships between solutions during historical environment changes. However, the complex relationships between solutions and the limited amount of available data often make it difficult to extract useful information efficiently, which may deteriorate the prediction accuracy. To address this problem, this article proposes a spatial–temporal topological tensor-based prediction method to generate the initial population in a new environment under the decomposition framework of MOEA/D. The method relies on the idea that the population distribution in each environment has topological similarity along the time dimension in the objective space, which makes it efficient to represent the population distribution in terms of a tensor and predict new solutions along each decomposition axis in a new environment by an improved tensor-based multishort time series prediction method. Experimental results on various benchmark problems and a real-world problem show that the proposed method is competitive or even superior to state-of-the-art dynamic multiobjective EAs based on prediction strategies.**

***Index Terms*—Dynamic multiobjective optimization, evolutionary algorithms (EAs), MOEA/D, topological tensor.**

Manuscript received 3 August 2023; revised 2 January 2024; accepted 11 February 2024. Date of publication 20 February 2024; date of current version 2 June 2025. This work was supported in part by the Major Program of National Natural Science Foundation of China under Grant 72192830 and Grant 72192831; in part by the Fund for the National Natural Science Foundation of China under Grant 62073067; and in part by the 111 Project under Grant B16009. This article was approved by Associate Editor J. Branke. *(Corresponding author: Lixin Tang.)*

Xianpeng Wang is with the National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Key Laboratory of Data Analytics and Optimization for Smart Industry, Ministry of Education, Northeastern University, Shenyang 110819, China (e-mail: wangxianpeng@ise.neu.edu.cn).

Yumeng Zhao is with the Liaoning Engineering Laboratory of Data Analytics and Optimization for Smart Industry and the Liaoning Key Laboratory of Manufacturing System and Logistics Optimization, Northeastern University, Shenyang 110819, China (e-mail: zhaoym@stumail.neu.edu.cn).

Lixin Tang is with National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University, Shenyang 110819, China (e-mail: lixintang@mail.neu.edu.cn).

Xin Yao is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong, China, and also with the School of Computer Science, University of Birmingham, Birmingham, U.K. (e-mail: xinyao@LN.edu.hk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TEVC.2024.3367747, provided by the authors.

Digital Object Identifier 10.1109/TEVC.2024.3367747

I. Introduction

DYNAMIC multiobjective problems (DMOPs) are optimization problems that contain multiple conflicting objective functions, constraints, or decision variables that change with time or environment [1]. There are many kinds of DMOPs in the real world, including raw ore allocation problem in mineral processing [2], path planning [3], [4], and process operation and scheduling problems in iron and steel industry [5], [6]. Due to constant changes in environments, it is a very challenging task to efficiently solve DMOPs [7]. Therefore, the study of dynamic multiobjective evolutionary algorithms (DMOEAs) for DMOPs has received much attention in recent years [8].

Since DMOP can be viewed as consisting of a series of static MOPs in a time period, the evolutionary algorithms (EAs) for the static MOP problem can be used to solve the DMOPs with appropriate modifications [9], [10]. To make static EAs better adapted to DMOPs, traditional methods focus on two key issues: 1) the ability of population to track dynamic environmental changes and 2) the ability to maintain population diversity whenever the environment changes [11]. Among these methods, the most popular approach in recent years is the prediction-based approach. This approach usually constructs a prediction model based on stored population or environmental information in different environments, and then uses this prediction model to predict the location of the Pareto optimal solutions in the new environment after a change in the environment [12]. There have been many prediction models proposed in the DMOEAs based on machine learning methods, such as neural networks [13], support vector machine [14], [15], clustering and correlation analysis [16], [17], transfer learning [12], [18], [19], and inverse Gaussian process (IGP) [20].

The machine learning-based prediction methods generally treat the generation of new population during environment changes as a "black box," which requires a large amount of data and a relatively time-consuming training process to ensure the accuracy of predictions. However, the dynamics of populations in response to environmental changes may not be an absolute "black box." For dynamic optimization problems in real production, the similarity in composition and size of successively processed jobs in the same production line makes the shapes of the Pareto fronts in different environments

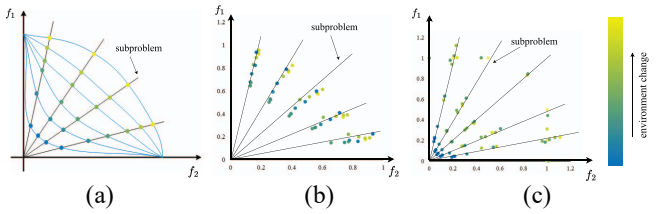


Fig. 1. Illustration of optimal Pareto front and practical evolutionary trajectory of solutions for subproblems under the decomposition framework of MOEA/D. (a) Distribution of Pareto front in each environment under MOEA/D. (b) Evolutionary process of subproblems of MOEA/D for DF3. (c) Evolutionary process of subproblems of MOEA/D for DF6.

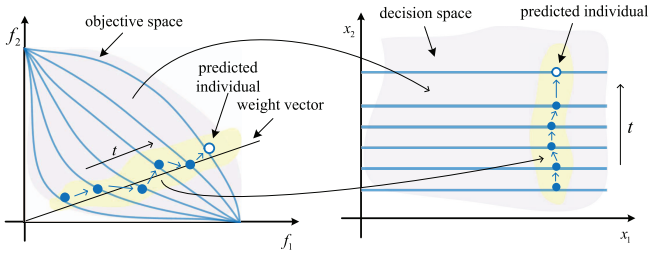


Fig. 2. Prediction of the initial solution for the new environment based on the evolutionary trajectory of solutions obtained by each subproblem.

rather similar [21]. For example, in the continuous annealing production process of an iron and steel company, many strips are welded one by one to be processed continuously. During practical production, once changes of the composition and process constraints of adjacent strips or changes of real-time production environment parameters, such as cooling gas temperature and quench water temperature, are detected, the control system needs to develop a new operation optimization scheme accordingly. So the operational optimization of the continuous annealing process can be considered as a dynamic multiobjective optimization problem. In fact, since the composition and size of the strips, as well as the optimization objectives in the operation optimization, are similar in nature, the shapes of the final evolved populations in each dynamic environment may also be similar.

As shown in Fig. 1(a), when dealing with DMOP under the decomposition framework of MOEA/D [22], the relative position relationship of optimal solutions among subproblems will remain unchanged with the change of environment because the optimal solutions of each subproblem in different environments will be distributed along the corresponding weight vector. To further verify this observation, evolutionary processes of some subproblems of MOEA/D for benchmark problems DF3 and DF6 are illustrated in Fig. 1(b) and (c), respectively. From the evolutionary results, it can be found that the best solutions obtained by each subproblem in different environments can constitute a relatively stable time series (TS). So, how to make use of the TS in the objective space to guide the generation of new individuals in the decision space for each subproblem in the new environment to ensure the diversity and convergence of the new population (Fig. 2), is a very challenging and important issue to be investigated.

To address the above issue, considering that tensor has a strong ability to characterize changes of spatial–temporal features of TS [23], we propose to use the spatial–temporal tensor (STT) to represent the dynamic changes of the final evolved populations along the different environments because the final evolved populations can be modeled as a high-order tensor. For example, for a bi-objective dynamic optimization problem, the final population obtained in each environment can be considered as a second-order tensor in which each fiber is a solution of a subproblem, and thus all populations obtained along the environment changes will constitute a third-order tensor (the example is given and explained by Fig. 3 in Section III-A). Based on the STT representation, we propose a dynamic MOEA/D with STT prediction (STT-DMOEA/D) for DMOPs in this article, whose main contributions are as follows.

1) Based on the decomposition framework of MOEA/D, the STT is adopted to explore the information and patterns of population changes, which in turn is used to guide the generation of initial populations that can better adapt to new environments. To the best of our knowledge, this is the first attempt to predict and generate the initial population in a new environment using STT, which has a good representation ability of topology structure and spatial–temporal changes of populations.
2) Based on the STT representation and the relatively stable trajectory of the best solutions found by each subproblem in different environments, an improved Block Hankel Tensor autoregressive integrated moving average (BHT-ARIMA) method is developed to generate the initial population in a new environment, which needs less training data and has a high-prediction accuracy for short-term trends.
3) Experimental results based on both benchmark DMOPs and real-world problems show that the proposed STT-DMOEA/D outperforms state-of-the-art DMOEAs based on prediction methods.

The remainder of this article is organized as follows. In Section II, definitions of DMOP and tensor are presented, followed by the review of related work in DMOEAs based on prediction methods. Section III describes the proposed STT-DMOEA/D algorithm in detail. The experimental results on both benchmark DMOPs and the real-world problem are provided and analyzed in Section IV. Finally, conclusions and future work are discussed in Section V.

II. Preliminaries and Related Work

A. Basic Definition of DMOPs

The unconstrained DMOP can be defined as follows [24]:

$$\begin{aligned} &\min F(x,t) = (f_1(x,t), f_2(x,t), \ldots, f_M(x,t))^T \\ &\text{s.t. } x \in \Omega, \quad t \in \Omega_t \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_v)$ is a v-dimension decision vector, $\Omega \subseteq R^v$ denotes the decision space, t is the discrete time, and $\Omega_t \subseteq R$ is the time space. $F(x,t) : (\Omega \times \Omega_t) \subseteq R^M$ is an objective vector consisting of M objectives changing with time, where R^M is the objective space.

B. Tensor and Tucker Decomposition

In this article, multidimensional data arrays are defined as a tensor, and the tensor dimensions are expressed by its order. The third-order tensor is mainly described and denoted by $\mathcal{X}$, and the tensor of dimension $I \times J \times K$ is denoted by $\mathcal{X}_{IJK}$ with subscripts in uppercase letters. The elements in $\mathcal{X}_{IJK}$ are denoted by $\mathcal{X}_{ijk}$ with subscripts in lower-case letters. The fibers of tensor $\mathcal{X}$ are the vectors extracted by fixing two of the dimensions, denoted by the lower-case letter x, while the slice of tensor $\mathcal{X}$ is the matrix extracted by fixing one of the dimensions, denoted by $\mathcal{X}_{i::}$, $\mathcal{X}_{:j:}$ and $\mathcal{X}_{::k}$, respectively.

Tucker decomposition [25], also known as higher-order singular value decomposition, is the decomposition of a tensor into the product of the core tensor and the corresponding matrix in each dimension, which can also be regarded as a higher-order version of principal component analysis. Taking the third-order tensor $\mathcal{X}_{IJK}$ as an example, its Tucker decomposition is

$$\begin{aligned}\mathcal{X}_{IJK} &\approx \mathcal{G} \times_1 A \times_2 B \times_3 C \\ &= \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}(a_p \circ b_q \circ c_r)\end{aligned} \tag{2}$$

where $A \in R^{I\times P}$, $B \in R^{J\times Q}$, $C \in R^{K\times R}$ are factor matrices in different dimensions. These matrices are usually considered as the principal components in different dimensions. $\mathcal{G} \in R^{P\times Q\times R}$ is called the kernel tensor, where each element represents the degree of interaction between different components, and can also be viewed as a compressed version of $\mathcal{X}_{IJK}$.

C. Related Work

DMOEAs proposed in [8] can be classified into three categories based on their strategies to extend static MOEAs to DMOPs: 1) diversity-based; 2) multipopulation-based; and 3) prediction-based approaches.

Diversity-based approaches usually use some strategy [26], [27], [28] to generate new individuals to maintain diversity, and this type of approach is more suitable for DMOPs with weak environmental changes. Multiple population-based methods are more likely to divide the population into multiple groups and ensure convergence and diversity of the population through co-operative [29] or separate [30] evolution. This type of approach is more effective for multipeaked DMOPs [12].

In recent years, prediction-based methods have attracted the attention of an increasing number of researchers. Such methods mainly utilize historical information to predict the locations of POS in a new environment. Many researchers construct prediction models that can learn from past experience to predict future changes [31], [32], [33]. To reduce the learning time, Muruganantham et al. [34] used a linear discrete-time Kalman filter model to predict the optimal values of decision variables in a new environment and proposed a dynamic MOEA/D algorithm. To predict more accurately the moving positions of solutions under multiple types of environment variations, two prediction approaches based on multimodel were proposed by Rong et al. [35], [36]. To improve the stability of multimodel prediction methods, a mixture-of-experts prediction framework was designed in [37] and a new information-sharing strategy was proposed in [38].

To achieve rapid response to environmental changes, Wang et al. [39] proposed a model based on gray prediction method and Zheng et al. [40] proposed a prediction method based on analysis and classification of decision variables. Considering that the successive environments were correlated, Cao et al. [41] used the centroids of historical POS to construct a difference model that could predict the movement trajectory of the centroids. A similar prediction method was proposed by Liu et al. [42] based on reference points and incorporated into a cooperative particle swarm optimization algorithm.

There are also many learning models that have been well applied to predict the initial populations of the new environment. For example, historical information was used to construct support vector regression (SVR) predictors in [14]. Zhang et al. [20] constructed an inverse Gaussian model by mapping from the objective space to the decision space. Xu et al. [15] constructed an incremental support vector machine by continuously accumulating historical information to realize the online prediction.

The transfer learning technique was used by Jiang et al. to generate the new initial population by reusing experience to speed up the evolutionary process, and Li et al. [12] further incorporated clustering into the transfer learning to reduce negative information transfer. Recently, Yu et al. [17] divided the population into three different clusters based on the correlation analysis of moving directions of individuals, and constructed different prediction models for each cluster. Finally, a hierarchical prediction method was constructed through the integration of these prediction models, and had shown very promising performance. Yu et al. [43] designed a historical evolution learning-based framework, in which two new models were proposed to assist the static optimizers. The proposed framework is very flexible and provided a new perspective for the using of historical evolution information.

Existing prediction-based methods, while mostly performing well, still have some shortcomings. First, the effectiveness of prediction can be compromised if there is not enough historical information to predict environmental changes in advance. Second, some prediction models are overly complex, resulting in algorithms that take too long to run. Third, each model that predicts solutions in a new environment is usually more applicable to a certain class of problems, and the convergence will be slowed down when the problem exhibits nonlinearity or insignificant regularity. The algorithm designed in this article will focus on overcoming and solving these issues.

III. Proposed Algorithm

In this section, we first describe the construction of the spatial–temporal topological tensor to represent the population changes in dynamic environments, and then present an improved version of BHT-ARIMA [44] that extends BHT-ARIMA to DMOPs with the STT representation of populations. After that, the STT-based generation method of initial population in the new environment (denoted as STT generator) is described based on the improved BHT-ARIMA

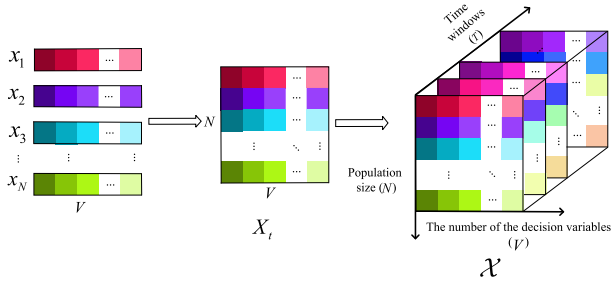


Fig. 3. Construction of the three-order STT based on final populations.

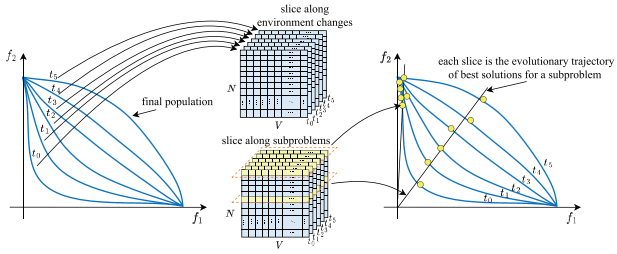


Fig. 4. Relationship between the STT and DMOEA.

method. Finally, the overall framework of the proposed STT-MOEA/D is presented and its computational time complexity is discussed.

A. Construction of Spatial–Temporal Topological Tensor

In our algorithm, the final populations along environment changes are represented by a three-order STT model $\mathcal{X} \in \mathcal{R}^{N\times V\times T}$, where N is the population size, V is the dimension of decision variables, and T is the number of environments. The construction of the tensor can be illustrated by Fig. 3. First, for each environment, an individual $x_n = \mathcal{X}_{n:t}$ $(n = 1, 2, \ldots, N)$ with V decision variables in the final population of environment t can be considered as a fiber of the tensor, and then all the N fibers (i.e., the final population in environment t) can be spelled into an $N \times V$ matrix X_t, which can be considered as a slice of the tensor, i.e., $X_t = \mathcal{X}_{::t} \in \mathcal{R}^{N\times V}$. As environment changes, the TS of slices can finally form a three-order tensor $\mathcal{X} \in \mathcal{R}^{N\times V\times T}$. Please note that the tensor is dynamic because a new final population will be added once the environment changes, so it is a three-order STT.

As shown in Fig. 4, if we cut the STT $\mathcal{X} \in \mathcal{R}^{N\times V\times T}$ vertically along the environment number dimension, we can get a slice $\mathcal{X}_{::t} \in \mathcal{R}^{V\times N}$, which is the final population in environment t. That is, the STT $\mathcal{X}$ has the representation ability of the dynamic distribution of final populations along environment changes. If we cut the STT $\mathcal{X} \in \mathcal{R}^{N\times V\times T}$ horizontally along the population size dimension, we will get a spatial–temporal slice $\mathcal{X}_{n::} \in \mathcal{R}^{V\times T}$, which represents the evolutionary trajectory of the best individuals obtained by each subproblem of MOEA/D along the environment changes. As analyzed before in Fig. 1, under the decomposition framework of MOEA/D, each spatial–temporal slice $\mathcal{X}_{n::}$ can be regarded as a relatively stable TS, and thus existing powerful TS analysis method, such as tensor-based autoregressive integrated moving average (ARIMA), can be used to predict the offspring solution for the new environment (Fig. 2). Since these new solutions are generated independently by mining useful information from the evolutionary trajectory of the best solutions obtained by each subproblem, they are generally of good quality and diversity, which helps to provide a high-quality initial population for the new environment. That is, the STT representation is suitable for modeling dynamics of final populations of the DMOEA, while the prediction of new initial population can accelerate the convergence of the DMOEA.

B. Improved BHT-ARIMA for DMOPs

Although the TS of $\mathcal{X}_{::1}, \mathcal{X}_{::2}, \ldots, \mathcal{X}_{::T}$ can be used to predict the next slice $\mathcal{X}_{::T+1}$ (i.e., the initial population in the new environment $T + 1$) with tensor-based ARIMA, the computational time complexity will be very high and the relative position relationship among solutions in each X_t cannot be fully used. To solve this problem, we take MOEA/D as the baseline algorithm and the weight vectors in MOEA/D unchanged. Based on this framework, for each weight vector (i.e., subproblem), we only need to predict a new individual based on a spatial–temporal slice P (i.e., the trajectory of the best individuals achieved by each subproblem), instead of predicting a whole new population based on $\mathcal{X}_{::1}, \mathcal{X}_{::2}, \ldots, \mathcal{X}_{::T}$. Since the prediction is performed only for a subproblem, the computational time complexity can be much reduced and the prediction quality of the new solution may be better. In the following, we will extend the original BHT-ARIMA to DMOPs based on the spatial–temporal slice $\mathcal{X}_{n::}$ and improve it by incorporating a dynamic parameter control strategy based on the stability definition of the spatial–temporal slice $\mathcal{X}_{n::}$ in the objective space.

1) Application of BHT-ARIMA to DMOPs: BHT-ARIMA is a new prediction model that works best for short TS and stationary series [45]. It first projects the kernel tensor after the multiway delay embedding transform (MDT) [46] and tucker decomposition, and then adds the generalized tensor ARIMA [47] to project future samples of the continuous kernel tensors. The BHT-ARIMA model has three advantages when handling DMOPs: 1) the tensor constructed from the processed historical data preserves the maximum continuity between environments; 2) the projected kernel tensor can better preserve the spatial correlation between the data, allowing for better capture of the intrinsic relationships between the data when constructing predictive models; and 3) good robustness can be obtained with reduced sensitivity to parameters. The flowchart of the extension of BHT-ARIMA to DMOPs (denoted as D-BHT-ARIMA) is shown in Fig. 5, and its process can be described as follows.

First, each slice $\mathcal{X}_{n::} \in \mathcal{R}^{V\times T}$ (i.e., the strategy of the nth subproblem) is transformed into a higher-order BHT $\hat{P}$ that is easier to learn and train using MDT along the time dimension

$$\hat{P} = \text{MDT}(\mathcal{X}_{n::}) \in \mathcal{R}^{V\times\tau\times(T-\tau+1)} \tag{3}$$

where τ is the parameter of MDT.

Second, since BHT-ARIMA predicts best for stationary series, $\hat{P}$ is differentiated to order d to obtain a stationary series $\Delta^d\hat{P} = \{\Delta^d\hat{P}_1, \Delta^d\hat{P}_2, \ldots, \Delta^d\hat{P}_k, \ldots, \Delta^d\hat{P}_K\} \in \mathcal{R}^{V\times\tau\times K}$, where $K = (T - \tau + 1 - d)$.

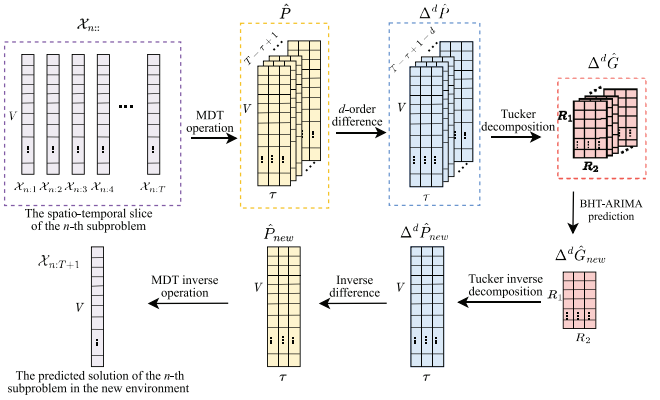


Fig. 5. Generation of an initial solution with D-BHT-ARIMA for the nth subproblem in the new environment $T+1$.

Third, each slice $\Delta^d\hat{P}_k$ in $\Delta^d\hat{P}$ is decomposed using the Tucker decomposition technique to obtain the kernel tensor:

$$\Delta^d\hat{G}_k = \Delta^d\hat{P}_k \times_1 \hat{U}^{(1)} \times_2 \hat{U}^{(2)} \tag{4}$$

where $\hat{U}^{(1)} \in \mathcal{R}^{V\times R_1}$ and $\hat{U}^{(2)} \in \mathcal{R}^{\tau\times R_2}$ are the factor matrices in two dimensions. Since the kernel tensor $\Delta^d\hat{G}_k \in \mathcal{R}^{R_1\times R_2}$ contains the most important information of the original BHT $\hat{P}$, and also reflects the internal interaction within the TS, it can be directly used to train the prediction model. Due to the fact that conventional ARIMA can only predict one scalar, modifications are made to it so as to predict multiple scalars, and the modified ARIMA prediction kernel tensor of order (p, d, q) is as follows [45]:

$$\Delta^d\hat{G}_k = \sum_{i=1}^{p}\alpha_i\Delta^d\hat{G}_{k-i} - \sum_{i=1}^{q}\beta_i\hat{\varepsilon}_{k-i} + \hat{\varepsilon}_k \tag{5}$$

where $\{\alpha_i\}_{i=1}^{p}$ and $\{\beta_i\}_{i=1}^{q}$ are the coefficients of AR and MA, respectively. They are estimated by following the classical ARIMA based on the Yule-Walker method [48]. $\hat{\varepsilon}_{k-i}$ is the random error of past q observations, and $\hat{\varepsilon}_k$ is the forecast error at the current time point, which should be minimized.

Fourth, by optimizing the factor matrix and the errors, a prediction model can be trained when the convergence condition is reached. The obtained model is then used to compute a new kernel tensor as

$$\Delta^d\hat{G}_{K+1} = \sum_{i=1}^{p}\alpha_i\Delta^d\hat{G}_{K-i} - \sum_{i=1}^{q}\beta_i\hat{\varepsilon}_{K-i}. \tag{6}$$

Fifth, the predicted kernel tensor is recovered using the Tucker inversion based on the optimized factor matrices $\hat{U}^{(1)}$ and $\hat{U}^{(2)}$

$$\Delta^d\hat{P}_{K+1} = \Delta^d\hat{G}_{K+1}\times_1\hat{U}^{(1)}\times_2\hat{U}^{(2)}. \tag{7}$$

Finally, the predicted values of all TS are obtained by performing the d-order inverse differentiation on $\Delta^d\hat{P}_{K+1}$, and thus the new BHT $\hat{P}_{\text{new}} \in \mathcal{R}^{V\times\tau\times(T-\tau+2)}$ for the new environment can be obtained. Then the MDT inverse transformation of $\hat{P}_{\text{new}}$ is performed to predict the new solution for the nth subproblem in environment $T+1$, as shown in (8).

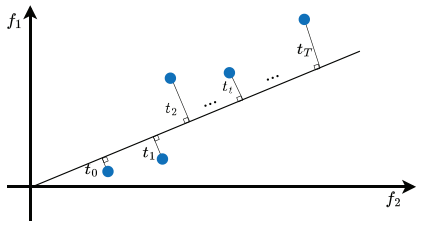


Fig. 6. Individual to weight vector distance in objective space.

Based on these new solutions, the initial population in the new environment $T+1$ can be obtained

$$\mathcal{X}_{n:T+1} = \text{MDT}^{-1}\left(\hat{P}_{\text{new}}\right) \in \mathcal{R}^{V\times(T+1)}. \tag{8}$$

2) Adaptive Selection of Differential Parameter d: As mentioned above, BHT-ARIMA works best for stable data and has three adjustable parameters—p, q, and d. Parameter p denotes the autoregressive order that reflects the consideration of past observations in the model, while parameter q represents the moving average order that indicates the consideration of past prediction errors. Parameter d is the differencing order, which denotes the number of differentiations required to achieve stationarity in the TS. Experimental results in [45] showed that parameters p and q have little effect on BHT-ARIMA, while the effect of parameter d is more pronounced. In the original BHT-ARIMA, the differential parameter d is set to a fixed value of 2. That is, each input TS needs to be differentiated twice, which is time-consuming. Therefore, in this article we propose an adaptive selection strategy of parameter d based on the stability of $\mathcal{X}_{n::}$. However, how to evaluate the stability of $\mathcal{X}_{n::}$ in the decision space is quite challenging, which is also the main reason that a fixed value of d was adopted in the original BHT-ARIMA. To handle this problem, we propose a new stability definition of $\mathcal{X}_{n::}$ in the objective space instead of the decision space.

As shown in Fig. 6, the stability of $\mathcal{X}_{n::}$ for the nth subproblem (weight vector) in the objective space is evaluated by the distance of each fiber (i.e., individual) in $\mathcal{X}_{n::}$ to the weight vector, and the adaptive selection strategy of the differential parameter d is determined as follows. First, calculate the distance of each fiber (i.e., individual $\mathcal{X}_{n:t}$) in $\mathcal{X}_{n::}$ to the weight vector, denoted as r_t. Second, construct the TS of distance $\Delta^0R = \{r_1, r_2, \ldots, r_t, \ldots, r_T\}$. Third, use the augmented Dickey–Fuller (ADF) [49] test to evaluate the stability of Δ^0R. If the ADF test result $\text{ADF}(\Delta^0R) \le 0.05$, then $d = 0$ is adopted. Otherwise, we further calculate the 1-order difference sequence of TS of distance, i.e., $\Delta^1R = \{r_2 - r_1, r_3, -r_2, \ldots, r_t - r_{t-1}, \ldots, r_T - r_{T-1}\}$. If the ADF test result $\text{ADF}(\Delta^1R) \le 0.05$, d is set to 1; otherwise, $d = 2$ is adopted. The selection of the differential parameter d is given in (9), based on which it is clear that $d = 2$ is adopted only in the worst case

$$d = \begin{cases} 0, & \text{if } \text{ADF}\left(\Delta^0R\right) \le 0.05 \\ 1, & \text{if } \text{ADF}\left(\Delta^0R\right) > 0.05 \ \&\ \text{ADF}\left(\Delta^1R\right) \le 0.05 \\ 2, & \text{otherwise.} \end{cases} \tag{9}$$

Algorithm 1: Initial Population Generation With STT

Input: All the final populations in previous T environments: POS_{all}; minimum size for STT model training: $Train_{low}$; weight vector of MOEA/D: $W = \{W_1, ..., W_N\}$; neighborhood size of MOEA/D: NER; mutation parameter: b

Output: Initial population in the new environment: $\mathcal{X}_{::T+1} \in \mathcal{R}^{N\times V}$;

1 $\mathcal{X} \in \mathcal{R}^{N\times V\times T} \leftarrow$ Construct STT tensor with POS_{all} ;

`// Data Complementation`

2 $\mathcal{X}^* =$ Complementation$(\mathcal{X}, Train_{low})$;

3 **for** $\mathcal{X}_{n::}$ *in* $\mathcal{X}^*$ **do**

`// Gaussian Variation`

4 **if** *non-singular decomposition issue occurs for D-BHT-ARIMA with* $\mathcal{X}_{n::}$ **then**

5 $ner =$Neighbor$(\mathcal{X}^*, NER, n)$;

6 $\mu =$Average(ner);

7 $\sigma =$StandardDeviation(ner);

8 $\mathcal{X}_{n::} =$GaussMutation(μ, σ, b);

9 $\mathcal{X}_{n::} =$BoundaryCheck$(\mathcal{X}_{n::})$;

10 $\mathcal{X}_{n:T+1} =$D-BHT-ARIMA$(\mathcal{X}_{n::})$;

11 $\mathcal{X}_{n:T+1} =$BoundaryCheck$(\mathcal{X}_{n:T+1})$;

12 $\mathcal{X}_{n:T+1} =$WeightSelection$(\mathcal{X}_{n:T+1}, \mathcal{X}_{n:T}, W_n)$;

13 **return** $\mathcal{X}_{::T+1} = \{\mathcal{X}_{1:T+1}, \mathcal{X}_{2:T+1}, \ldots, \mathcal{X}_{N::T+1}\}$;

C. Generation of Initial Population With STT

Based on D-BHT-ARIMA, the generation procedure of initial population with STT generator is given in Algorithm 1. When the environment changes, there are still two issues to deal with in the application of D-BHT-ARIMA. One is that the prediction of new solutions with D-BHT-ARIMA requires the accumulation of some historical data. At the early stage of environmental changes, data in the temporal dimension of the STT $\mathcal{X} \in \mathcal{R}^{N\times V\times T}$ is often insufficient. The other problem is that when the data in the temporal slice P is not a low-rank matrix, there is a situation that the internally computed matrix cannot be decomposed properly for nonsingularity. Focusing on the two issues, two strategies are developed, and a weight vector-based selection strategy to refine the predicted population is also adopted based on the decomposition framework of MOEA/D.

1) Data Complementation Strategy: To deal with the issue of insufficient historical training data, a complementation operation is performed on the STT $\mathcal{X} \in \mathcal{R}^{N\times V\times T}$ (Algorithm 1, line 2). When the length of $\mathcal{X} \in \mathcal{R}^{N\times V\times T}$ (i.e., T) is smaller than the minimum size of training data (denoted as $\text{Train}_{\text{low}}$), we split $\mathcal{X}$ into T slices $\mathcal{X}_{::1}, \mathcal{X}_{::2}, \ldots, \mathcal{X}_{::T}$ of $N\times V$ along the time dimension, and build T sets of archives, each of which contains a slice. All the slices will be copied in each archive until the sum of slices exceeds $\text{Train}_{\text{low}}$, and then they will be merged one by one to construct a new STT $\mathcal{X}^*$, which is used as the training data set. An example of the complementation operation with $T = 4$ and $\text{Train}_{\text{low}} = 9$ is shown in Fig. 7.

2) Gaussian Variation Strategy: To handle the nonsingular decomposition issue of internal matrixes and ensure that the spatial–temporal slice $\mathcal{X}_{n::}$ can be computed by D-BHT-ARIMA, we apply the Gaussian mutation to the complemented data (Algorithm 1, lines 5–9). More specifically, when a spatial–temporal slice cannot be computed by the STT generator, we first compute the mean and variance of the slices within the neighborhood of the weight vector to which this spatial–temporal slice belongs. Next, a number of b fibers of this slice are randomly selected for Gaussian mutation and then the boundary check will be performed to ensure the feasibility of variables. Through this operation, the mutated slice $\mathcal{X}_{n::}$ can be computed by D-BHT-ARIMA.

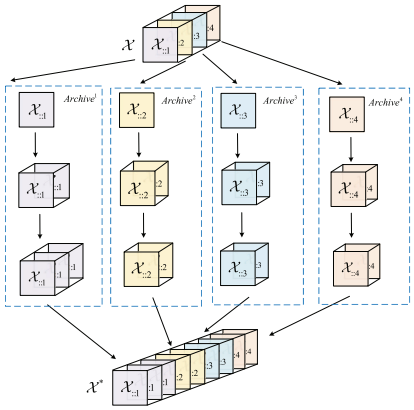


Fig. 7. Complementation operation with $K = 4$ and $Train_{low} = 9$.

3) Weight Vector-Based Selection Strategy: After a new individual $\mathcal{X}_{n:T+1}$ is generated for the nth weight vector (subproblem) by the prediction model of D-BHT-ARIMA and the boundary check process (Algorithm 1, lines 10 and 11), it will be compared with the individual $\mathcal{X}_{n:T}$ in the previous environment T. The comparison is based on the Chebyshev aggregation method (the evaluation of $\mathcal{X}_{n:T}$ is illustrated in (10)), and the better one is selected as the initial solution of nth weight vector in the new environment $T + 1$

$$f(\mathcal{X}_{n:T}) = \max_{1\le i\le M}\left\{w_i\left(F_i(\mathcal{X}_{n:T}, T) - z_i^*\right)\right\} \tag{10}$$

where w_i is the weight for the ith objective and z_i^* represents the ith objective of the ideal point z^* of MOEA/D.

D. Framework of the Proposed Algorithm

Based on the above STT generator, the overall framework of STT-DMOEA/D for DMOPs is presented in Algorithm 2. For each environment T, the static MOEA/D is performed to find the near optimal Pareto set POS_T and the corresponding Pareto front PF_T with the given initial population POP_T. Whenever the environment changes, the STT generator is used to generate the initial population for the new environment and then the static MOEA/D is performed to evolve it.

E. Computational Complexity

Once an environment change occurs, in the proposed STT-DMOEA/D algorithm, D-BHT-ARIMA will be used to generate the initial population and then MOEA/D is adopted to search for the new Pareto front based on the generated initial population. For the generation of a new solution for each

Algorithm 2: STT-DMOEA/D

Randomly initialize a population POP_1, set $T = 1$ and initialize the parameters used in D-BHT-ARIMA and MOEA/D;
$POS_0 = \mathrm{M}\,OEA/\mathrm{D}(POP_0,\ F(x,0),\ \ N, V)$;
$POS_{all} = POS_0$;
while *termination criterion is not reached* **do**
  if *environment changes* **then**
    $T = T + 1$;
    $POP_T = \mathrm{STT}(POS_{all})$;
  $POS_T = \mathrm{M}\,OEA/\mathrm{D}(POP_T)$;
  $POS_{all} = POS_{all} \cup POS_T$;
return POS_t;

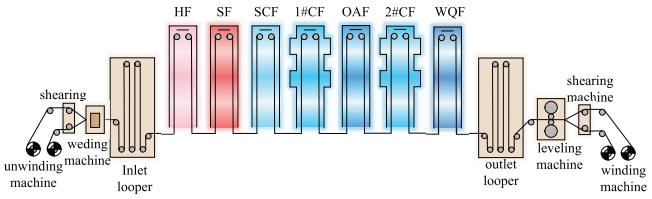


Fig. 8. Process of continuous annealing production.

subproblem, D-BHT-ARIMA consists of mainly MDT operation, Tucker decomposition and ARIMA prediction, whose computational complexities are $\mathrm{O}(VT^3)$, $\mathrm{O}(VTn^2)$, and $\mathrm{O}((p+q)T)$, where V and T are, respectively, the dimension of variables and the number of environmental changes, $n = 2$ is the order of Tucker decomposition, and $p = 1$ and $q = 3$ are the parameters of ARIMA. Due to the fact that there are a total of N subproblems, the computational complexity of D-BHT-ARIMA for generating the initial population of the new environment is $\mathrm{O}(NVT^3)$. Considering that the computational complexity of MOEA/D is $\mathrm{O}(mNB)$, where m is the number of objectives ($m = 2$) and B ($B = 20$) is the neighborhood size of each subproblem, the total computational complexity of our STT-DMOEA/D algorithm is $\mathrm{O}(NVT^3)$. That is, the time-consuming operation is mainly the D-BHT-ARIMA.

IV. Experimental Study

A. Test Problems

1) Benchmark Problems: The CEC 2018 DMO benchmark problems [50] are selected as test problems, which include classical dynamic multiobjective benchmark problems, such as FDA [51], DMOP [29], JY [52], and SDP [53]. This set of test problems consists of 14 benchmark test functions, namely, DF1-DF14, in which there are nine two-objective and five three-objective problems. These problems cover a wide range of problem features, such as irregular POF, discontinuous POF, and POS with time-varying geometries, which are a good representation of real-world situations.

2) Real World Problem: To verify the performance of STT-DMOEA/D in real world applications, the dynamic operation optimization of continuous annealing in the iron and steel industry is selected. As shown in Fig. 8, the welded strips will be consecutively processed through heating furnace (HF), soaking furnace (SF), slow cooling furnace (SCF), 1# fast cooling furnace (1#CF), over-aging furnace (OAF), 2# fast cooling furnace (2#CF), and water quenching furnace (WQF). In each furnace, a number of control variables need to be dynamically optimized [54]. In experiment, we take these process control variables of each strip as decision variables to simultaneously optimize three objectives, i.e., maximization of unit capacity, minimization of energy consumption, and minimization of quality difference between the practical and desired metrics. In addition, the temperature range of furnaces that can be dynamically adjusted for the adjacent strips is taken as the process constraints. The constraint handling method in NSGA-II [55] is adopted to deal with these process constraints in our problem. As mentioned in Section I, the fluctuations of the composition and process constraints of adjacent strips or the changes of real-time production environment parameters, such as cooling gas temperature and quench water temperature, are considered as environment changes. Since the true Pareto optimal sets are not known for this real-world problem, we prefer to use the nondominated solutions selected from the union of all approximate Pareto sets obtained by all the testing algorithms after multiple runs as the reference Pareto sets and Pareto fronts to compute the performance metrics. Due to page limitation, the decision variables and dynamic operation optimization model of the continuous annealing process are provided in the supplementary file.

B. Performance Metrics

In experiments, two commonly adopted performance metrics in DMOP, i.e., mean inverted generational distance (MIGD) [21], [34] and mean hypervolume (MHV) [56], are selected to evaluate the performance of algorithms.

1) MIGD: IGD can evaluate both convergence and diversity of an algorithm by calculating the difference between the true POF and the approximated POF achieved by the algorithm. A smaller value of IGD represents a better performance of the algorithm. The IGD metric for environment t ($t = 1, 2, \ldots, T$) is calculated as follows:

$$\mathrm{IGD}_t\left(\mathrm{POF}_t^*, \mathrm{POF}_t\right) = \frac{\sum_{p\in \mathrm{POF}_t^*} d(p, \mathrm{POF}_t)}{\left|\mathrm{POF}_t^*\right|} \tag{11}$$

where POF_t^* and POF_t are the true POF and the approximated Pareto front obtained by an algorithm in environment t, respectively. $d(p, \mathrm{POF}_t)$ represents the minimum Euclidean distance between solution p in POF_t^* and the solutions in POF_t. Based on IGD, the MIGD can be used to evaluate the performance of an algorithm for DMOP

$$\mathrm{MIGD} = \frac{\sum_{t\in T} \mathrm{IGD}\left(\mathrm{POF}_t^*, \mathrm{POF}_t\right)}{|T|}. \tag{12}$$

2) MHV: Hypervolume (HV) is another performance metric commonly used for MOEA, and a larger HV indicates a better performance of an algorithm. The MHV metric is calculated by (13), where HV_t represent the HV value of the Pareto front obtained in environment t

$$\mathrm{MHV} = \frac{\sum_{t\in T} \mathrm{HV}_t}{|T|}. \tag{13}$$

C. Compared Algorithms

To verify the performance of the proposed STT-DMOEA/D algorithm, four representative and relatively new DMOEAs were selected for comparison. These four algorithms use different prediction strategies to generate the initial populations in the new environment. For a fair comparison, we replaced the baseline algorithm with MOEA/D for all compared algorithms, which can be described as follows.

1) Tr-DMOEA [18]: The algorithm uses transfer learning for prediction and exploits the correlation of probability distributions of solutions in different environments to generate valid initial populations. This approach overcomes the problem of varying POFs in traditional machine learning-based prediction models leading to training and prediction samples with different distributions, and greatly improves the solution quality and the robustness of the algorithm.

2) IGP-DMOEA [20]: This algorithm uses the IGP to construct the predictor that can map POS from the objective space to the decision space. This method is able to obtain more convergent and better-diverse solutions in the objective space and is more responsive to decision makers than traditional methods.

3) ISVM-DMOEA [15]: This algorithm is a relatively new representative algorithm that combines machine learning with EAs. It treats DMOPs as an online learning process and updates the incremental support vector machine using successively obtained optimal solutions without discarding previous solution information. The algorithm can make better use of the nonlinear relationships between solutions and historical information to deal with DMOPs efficiently.

4) PPS [21]: This algorithm is a very classical DMOAs based on the prediction strategy. It combines a center point and a manifold to predict the initial population in the next time window, which is effective in handling DMOPs.

D. Parameter Settings

In the experiment, all parameters of the compared algorithms are taken from the corresponding original literature. The source codes of all the testing algorithms are available from the corresponding author upon reasonable request. Some parameters of STT-DMOEA/D are set as follows.

1) Basic Parameters: The population size is set to 300 for bi-objective problems (DF1-9) and 1326 for tri-objective problems (DF10-14). The number of decision variables is set as 10 for all test problems. The neighborhood size *NER* is 20. The crossover probability, variance probability and neighbor selection probability are 0.5, 0.5 and 0.8, respectively.

2) Dynamic Characteristics: In the test problems, time is defined as $t = (1/n_t)\lfloor(\tau_i/\tau_t)\rfloor$, where n_t, τ_t, and τ_i represent the severity of environmental changes, the frequency of environmental changes, and the number of iterations, respectively. Different combinations of n_t and τ_t represent different dynamic characteristics, and four combinations are used: 1) $(n_t = 5, \tau_t = 10)$; 2) $(n_t = 10, \tau_t = 5)$; 3) $(n_t = 10, \tau_t = 10)$; and 4) $(n_t = 10, \tau_t = 20)$. $\lfloor(\tau_i/\tau_t)\rfloor$ is fixed to 30 to ensure 30 environment changes in each problem. Before the first environment change, each algorithm is allowed to run for 50 generations, and each algorithm runs for 20 independent runs for each problem.

3) Parameters of STT Generator: In the STT generator, $\text{Train}_{\text{low}}$ of the tensor training model is set to 8, and the corresponding MDT rank decomposition parameter τ is 2. The random sampling parameter b in the Gaussian variant is set to 3. Because of the robustness of BHT-ARIMA, the parameters of ARIMA, i.e., q and p, are set to 3 and 1, respectively. For the kernel tensor parameters in the Tucker decomposition, R_1 is set to 3 and R_2 is set to the number of objectives.

4) Continuous Annealing Problem Parameters: After preprocessing, three sets of strip data were selected from practical production, and two sets of dynamic characteristics are used, i.e., $(1/n_t = 1, \tau_t = 10)$ and $(1/n_t = 1, \tau_t = 30)$. Before the first environment change, each algorithm is allowed to run for 150 generations.

E. Comparison Results and Analysis

1) Benchmark Problems: The mean and standard deviation of MIGD and MHV metrics obtained by the proposed STT-DMOEA/D and the other algorithms for different combinations of (τ_t, n_t) are given in Tables I and II, respectively. In the tables, the best result for each problem is marked with dark gray shading, and the second best result is marked with light gray shading. The Wilcoxon rank sum test [57] with the 0.05 significance level is carried out to indicate the significance between different results. The symbols "+", "=", and "−" are used to show that the proposed STT-DMOEA/D is significantly better than, not significantly different from, and significantly worse than a certainly compared algorithm.

Fig. 9 depicts the evolution curves of the average IGD values of DF1-DF14 over 20 independent runs along 30 environmental changes with $\tau_t = 10$ and $n_t = 10$, and the POF diagrams obtained by different algorithms for problems DF1-DF9 are shown in Fig. 1 in the supplementary file.

As can be seen in Table I, STT-DMOEA/D obtained the best MIGD for most of the benchmark problems. For the bi-objective problems DF1-DF9, STT-DMOEA/D achieved the best-MIGD results for 25 out of the 36 instances, and the second best-MIGD results for six instances. For the MHV metric, its performance is similar to others. In the case of DF2 and DF3, where STT-DMOEA/D performs slightly worse, DF2 is a slightly more complex problem whose variation leads to a loss of population diversity, and DF3 has little dynamic variation and mainly assesses the algorithm's ability to track concave and convex changes. From the evolution process of MIGD for DF3 shown in Fig. 8, it is clear that the proposed STT-DMOEA/D is inferior to IGP-DMOEA/D at the early environments but gradually becomes the best one for later environments. The main reason behind this phenomenon is that in the early environments, the available historical information is relatively insufficient for the STT generator to predict good new populations, however, as more historical training data becomes available, the prediction quality of the STT generator becomes better, which helps the proposed STT-DMOEA/D to achieve the best-IGD metrics in the later environments.

TABLE I
MEAN AND STANDARD DEVIATION VALUES OF MIGD OBTAINED BY STT-DMOEA/D AND COMPARED ALGORITHMS

Prob.	(τ_t, n_t)	Tr-DMOEA/D	IGP-DMOEA/D	ISVM-DMOEA/D	PPS-DMOEA/D	STT-DMOEA/D
DF1	(10,5)	6.5352e-03(1.18e-04)-	8.8003e-03(6.51e-03)-	1.3622e-02(1.19e-02)+	2.8791e-02(4.02e-01)+	1.0833e-02(9.08e-03)
	(5,10)	1.8598e-02(1.53e-03)+	1.6701e-02(8.78e-03)+	6.8655e-02(1.21e-02)+	1.0407e-01(3.17e-02)+	1.4551e-02(1.81e-03)
	(10,10)	5.5807e-03(2.71e-03)+	6.1655e-03(6.90e-04)+	1.3128e-02(1.08e-03)+	2.7769e-02(1.64e-02)+	4.6126e-03(7.76e-04)
	(20,10)	2.7529e-03(7.33e-04)+	2.7008e-03(1.87e-04)+	8.6469e-03(2.31e-04)+	3.4186e-03(2.70e-03)+	2.1815e-03(5.28e-04)
DF2	(10,5)	1.7301e-02(2.29e-02)-	1.1533e-02(1.05e-02)-	1.3595e-02(1.57e-03)-	2.8048e-02(2.29e-02)-	3.3579e-02(1.39e-02)
	(5,10)	1.3903e-02(3.04e-03)-	3.1766e-02(1.89e-03)+	9.8484e-02(1.16e-02)+	1.0061e-01(4.32e-02)+	4.5505e-02(1.62e-02)
	(10,10)	5.5114e-03(1.98e-02)-	8.8979e-03(6.58e-03)-	1.1527e-02(8.06e-02)-	2.7509e-02(2.39e-02)+	2.8219e-02(7.47e-03)
	(20,10)	2.7649e-03(2.58e-02)-	2.9400e-03(1.92e-04)-	8.9965e-03(4.56e-03)-	3.4307e-03(3.30e-03)+	1.1537e-02(4.87e-03)
DF3	(10,5)	1.9166e-01(1.06e-01)+	2.1082e-02(4.24e-03)-	2.2883e-01(1.80e-02)+	1.6210e-01(1.91e-02)+	5.7749e-02(1.83e-02)
	(5,10)	1.6294e-01(2.54e-02)+	4.0396e-02(6.65e-03)-	2.2755e-01(4.14e-02)+	1.2413e-01(1.43e-02)+	9.6375e-02(8.01e-02)
	(10,10)	9.9111e-02(1.02e-02)+	1.5768e-02(1.65e-03)-	2.0953e-01(1.37e-02)+	6.0635e-02(2.02e-02)+	4.4504e-02(2.71e-03)
	(20,10)	6.4484e-02(5.44e-03)+	6.7727e-03(1.87e-03)-	1.8252e-01(2.16e-02)+	2.5131e-02(2.24e-02)+	1.0992e-02(1.98e-02)
DF4	(10,5)	1.0900e-01(1.72e-03)=	1.0671e-01(4.68e-04)=	1.1624e-01(2.22e-03)+	1.1381e-01(1.01e-03)+	1.0296e-01(1.16e-03)
	(5,10)	1.5230e-01(2.45e-02)+	1.1230e-01(5.22e-03)+	3.4184e-01(3.86e-02)+	1.1556e-01(3.58e-02)+	1.0543e-01(5.62e-03)
	(10,10)	1.0557e-01(7.23e-02)+	1.0460e-01(1.31e-03)=	1.1420e-01(1.48e-02)+	1.0877e-01(1.72e-02)+	1.0047e-01(2.06e-04)
	(20,10)	1.0371e-01(3.95e-03)+	1.0271e-01(1.44e-03)=	1.1294e-01(6.38e-03)+	1.0260e-01(1.96e-02)=	1.0034e-01(2.75e-04)
DF5	(10,5)	4.4869e-03(2.85e-04)+	4.3958e-03(2.65e-04)+	5.7778e-03(5.69e-04)+	5.9151e-03(6.13e-04)+	4.0996e-03(1.11e-04)
	(5,10)	8.2254e-03(3.14e-03)+	7.8673e-03(1.89e-03)+	8.5704e-02(4.17e-02)+	2.2769e-02(5.33e-02)+	6.3993e-03(1.56e-03)
	(10,10)	3.6266e-03(2.97e-04)+	3.6578e-03(4.13e-04)+	4.9954e-03(3.14e-03)+	8.3959e-03(1.17e-02)+	3.1853e-03(3.28e-04)
	(20,10)	2.1838e-03(9.79e-02)+	2.2587e-03(8.60e-05)+	3.2830e-03(4.40e-04)+	2.3289e-03(2.47e-03)+	1.9596e-03(9.46e-05)
DF6	(10,5)	9.9875e-01(3.25e-01)+	3.2511e-01(1.72e-01)+	9.3789e-01(1.90e-01)+	8.9961e-01(1.09e-01)+	3.0141e-01(2.55e-01)
	(5,10)	1.0114e-00(2.01e-01)+	8.2470e-01(1.19e-00)+	2.3642e-00(3.27e-01)+	2.7971e-00(6.93e-01)+	6.4736e-01(7.63e-01)
	(10,10)	9.3491e-01(3.44e-01)+	3.1902e-01(8.02e-01)+	1.0472e-00(3.32e-01)+	8.9929e-01(4.35e-01)+	2.5550e-01(3.02e-01)
	(20,10)	6.0426e-01(3.29e-01)+	2.9406e-01(2.76e-01)+	8.1260e-01(3.59e-02)+	7.7379e-01(2.51e-01)+	2.3541e-01(2.84e-01)
DF7	(10,5)	8.4135e-02(3.02e-03)+	3.9658e-02(1.34e-03)+	4.4198e-01(8.02e-02)+	3.1342e-01(7.37e-02)+	1.3866e-02(2.04e-02)
	(5,10)	9.6858e-02(2.83e-03)+	5.4721e-02(8.33e-03)+	2.3254e-01(2.80e-02)+	6.7866e-02(2.70e-02)+	3.3943e-02(9.32e-04)
	(10,10)	7.4511e-02(3.63e-03)+	3.6626e-02(4.26e-03)+	1.3496e-01(2.58e-02)+	5.0646e-02(6.14e-03)+	7.7556e-03(4.35e-03)
	(20,10)	5.1598e-02(2.96e-04)+	8.0139e-03(4.36e-03)=	9.7671e-02(2.55e-02)+	3.5481e-02(1.65e-03)+	6.1392e-03(4.20e-03)
DF8	(10,5)	1.8563e-02(1.25e-02)+	1.0763e-02(1.66e-04)=	1.6575e-02(1.21e-03)+	2.5564e-02(1.67e-03)+	1.0741e-02(1.09e-02)
	(5,10)	2.3179e-02(4.91e-03)+	1.1544e-02(1.81e-03)=	4.2413e-02(2.69e-02)+	2.9431e-02(7.80e-03)+	1.1144e-02(5.76e-04)
	(10,10)	1.1736e-02(1.69e-03)+	9.0771e-03(9.12e-04)=	1.2943e-02(4.09e-03)+	2.4534e-02(2.91e-03)+	8.6190e-03(2.03e-04)
	(20,10)	9.1460e-03(2.59e-04)+	7.7875e-03(7.88e-04)=	1.0990e-02(5.96e-03)+	8.9607e-03(4.23e-03)+	8.0135e-03(2.18e-04)
DF9	(10,5)	4.2015e-02(4.32e-03)=	4.6244e-02(4.14e-03)=	1.2113e-01(1.52e-02)+	1.0914e-01(2.52e-02)+	4.5837e-02(1.25e-02)
	(5,10)	6.5623e-02(2.66e-02)+	5.2005e-02(8.18e-03)+	3.3758e-01(6.84e-02)+	1.8017e-01(9.64e-02)+	7.1625e-02(6.85e-02)
	(10,10)	2.7348e-02(5.71e-03)-	2.6603e-02(1.62e-02)=	1.1703e-01(2.38e-02)+	6.0939e-02(2.16e-02)+	3.4143e-02(2.76e-02)
	(20,10)	1.3935e-02(4.37e-03)=	1.6329e-02(7.04e-03)=	6.8254e-02(2.42e-02)+	1.9269e-02(2.39e-02)+	1.3045e-02(2.09e-02)
DF10	(10,5)	1.2539e-01(6.09e-03)+	1.1111e-01(1.15e-02)+	3.1315e-01(3.52e-02)+	8.8371e-02(7.21e-03)+	7.5637e-02(1.65e-03)
	(5,10)	1.6185e-01(1.75e-02)+	1.4850e-01(1.65e-02)+	3.7307e-01(6.80e-02)+	1.0192e-01(1.33e-02)+	8.8701e-02(2.09e-03)
	(10,10)	1.1753e-01(7.46e-03)+	1.0755e-01(5.08e-02)+	2.7102e-01(3.76e-02)+	8.3155e-02(1.65e-02)+	7.0279e-02(1.24e-03)
	(20,10)	1.1085e-01(7.44e-03)+	8.9591e-02(4.76e-03)+	2.4262e-01(6.18e-02)+	6.5747e-02(1.51e-03)=	6.3357e-02(8.21e-04)
DF11	(10,5)	2.9652e-01(1.52e-02)+	2.5771e-02(5.99e-05)+	6.9563e-02(1.85e-02)+	2.6583e-02(2.53e-03)+	2.1352e-02(1.32e-04)
	(5,10)	3.9403e-02(3.16e-03)+	2.7480e-02(3.36e-03)+	8.3376e-02(2.98e-02)+	2.9688e-02(4.03e-03)+	2.3122e-02(3.72e-04)
	(10,10)	2.7993e-02(1.97e-03)+	2.2462e-02(3.08e-04)+	6.0343e-02(1.67e-02)+	2.5073e-02(2.83e-04)=	2.0299e-02(1.42e-04)
	(20,10)	2.3913e-02(9.81e-04)+	1.9983e-02(1.58e-04)+	4.8293e-02(1.60e-02)+	2.2774e-02(2.90e-04)=	1.8864e-02(7.40e-05)
DF12	(10,5)	1.8563e-01(1.64e-02)+	1.1674e-01(1.03e-02)+	3.2465e-01(4.85e-03)+	2.5621e-01(1.53e-02)+	2.9536e-02(5.63e-03)
	(5,10)	2.0027e-01(6.94e-02)+	1.2166e-01(3.38e-03)+	4.1682e-01(8.00e-03)+	2.4000e-01(7.16e-03)+	3.3794e-02(2.84e-03)
	(10,10)	1.7679e-01(1.66e-02)+	9.1002e-02(8.04e-03)+	3.0810e-01(4.79e-03)+	2.2013e-01(1.51e-02)+	2.7880e-02(9.00e-04)
	(20,10)	1.1941e-01(5.49e-03)+	4.8061e-02(1.20e-03)+	2.9308e-01(5.69e-03)+	1.2158e-01(1.53e-02)+	2.3026e-02(1.64e-03)
DF13	(10,5)	1.5962e-02(4.62e-03)-	1.6504e-01(4.45e-03)=	1.4863e-01(1.46e-02)-	1.6001e-01(8.56e-03)-	1.6521e-01(2.52e-03)
	(5,10)	1.6437e-01(1.94e-02)-	1.6295e-01(1.35e-03)-	1.6035e-01(3.37e-03)-	1.6194e-01(8.73e-03)-	1.6634e-01(3.43e-03)
	(10,10)	1.5256e-01(5.51e-03)-	1.5855e-01(2.29e-03)-	1.4464e-01(1.72e-02)-	1.5995e-01(9.68e-03)-	1.6481e-01(1.85e-03)
	(20,10)	1.5043e-01(1.31e-03)-	1.5605e-01(1.80e-03)-	1.2049e-01(6.23e-03)-	1.5898e-01(4.08e-03)-	1.6235e-01(4.03e-03)
DF14	(10,5)	1.8062e-02(1.35e-03)+	2.0714e-02(4.61e-05)+	1.8569e-02(5.62e-02)+	1.8365e-02(1.85e-03)+	1.7586e-02(1.98e-02)
	(5,10)	2.2677e-02(1.61e-03)+	2.1201e-02(1.39e-03)+	3.8445e-02(1.43e-02)+	2.9397e-02(1.45e-02)+	1.9573e-02(1.21e-03)
	(10,10)	1.7935e-02(1.22e-03)+	1.7771e-02(1.86e-03)+	1.8460e-02(6.68e-03)+	1.7934e-02(1.93e-03)+	1.7262e-02(2.05e-02)
	(20,10)	1.6344e-02(1.20e-03)+	1.6385e-02(1.93e-03)+	1.6492e-02(1.38e-03)+	1.6298e-02(1.14e-03)=	1.6080e-02(8.94e-02)
1st/2nd/all		5/12/56	6/31/56	4/1/56	0/6/42	41/4/56

Among the 9 bi-objective problems, DF6 is a multimodal and difficult problem. For this problem, the proposed STT-DMOEA/D can achieve much better-MIGD results than all other rival algorithms, and as can be seen from the POF diagrams shown in Fig. 1 in the supplementary file, only the proposed STT-DMOEA/D can achieve good convergence to the true Pareto front in every environment, while other rival algorithms cannot handle this problem well.

For the tri-objective problems DF10-DF14, the STT-DMOEA/D algorithm can achieve the best-MIGD metrics for 12 out of the 15 instances (i.e., all problems except DF13). The main reason for the inferior performance of STT-DMOEA/D for problem DF13 is that DF13 has both continuous and disconnected POFs, and the disconnected POFs are time-varying. The relatively complex variation of decision variables in DF13 leads to poor predictions by the STT generator.

TABLE II
MEAN AND STANDARD DEVIATION VALUES OF MHV OBTAINED BY STT-DMOEA/D AND COMPARED ALGORITHMS

Prob.	(τ_t, n_t)	Tr-DMOEA/D	IGP-DMOEA/D	ISVM-DMOEA/D	PPS-DMOEA/D	STT-DMOEA/D
DF1	(10,5)	4.2062e-01(2.16e-04)-	4.2366e-01(9.67e-04)-	3.6218e-01(1.47e-02)+	3.0151e-01(3.30e-03)+	4.1484e-01(1.08e-02)
	(5,10)	4.1541e-01(3.18e-03)=	4.1255e-01(5.31e-03)=	3.0701e-01(7.89e-03)+	2.9771e-01(7.80e-03)+	4.1033e-01(1.33e-02)
	(10,10)	4.2253e-01(8.63e-04)=	4.3033e-01(8.22e-04)=	3.8660e-01(6.46e-03)+	3.1207e-01(1.26e-02)+	4.3324e-01(1.11e-03)
	(20,10)	4.3645e-01(2.33e-04)=	4.3653e-01(2.60e-04)=	4.2464e-01(3.53e-03)+	4.3507e-01(3.69e-03)+	4.3749e-01(2.08e-04)
DF2	(10,5)	6.4672e-01(6.45e-04)-	6.5001e-01(9.18e-04)-	6.4466e-01(2.03e-03)-	6.2683e-01(2.37e-03)-	6.1325e-01(1.24e-02)
	(5,10)	6.4356e-01(8.72e-04)-	6.1773e-01(8.93e-03)-	5.0123e-01(1.37e-02)+	5.6869e-01(1.63e-02)+	5.9678e-01(2.97e-02)
	(10,10)	6.4599e-01(6.07e-04)-	6.5202e-01(2.94e-03)-	6.1595e-01(5.49e-03)+	5.2341e-01(1.39e-02)+	6.3803e-01(1.22e-02)
	(20,10)	6.6215e-01(4.68e-04)=	6.6176e-01(3.07e-04)=	6.5217e-01(1.11e-03)=	6.6109e-01(3.79e-03)+	6.6088e-01(3.21e-03)
DF3	(10,5)	2.5319e-01(1.25e-03)+	3.6126e-01(3.47e-03)-	2.1521e-01(1.20e-02)+	2.5461e-01(1.06e-02)+	3.3359e-01(1.51e-02)
	(5,10)	2.4266e-01(1.47e-02)+	3.2866e-01(1.99e-02)+	9.1192e-02(1.22e-02)+	2.6641e-01(2.06e-02)+	3.1872e-01(1.56e-02)
	(10,10)	2.6773e-01(1.19e-03)+	3.7126e-01(1.06e-02)=	1.5314e-01(1.39e-02)+	2.2895e-01(1.49e-02)+	3.6978e-01(9.97e-03)
	(20,10)	2.9619e-01(5.44e-04)+	3.8969e-01(6.94e-03)=	1.9828e-01(9.70e-03)+	3.7334e-01(5.39e-03)+	3.8935e-01(5.11e-04)
DF4	(10,5)	7.0923e-00(3.82e-03)+	7.3199e-00(3.11e-03)=	7.0846e-00(4.81e-03)+	7.0152e-00(7.26e-04)+	7.3249e-00(3.53e-03)
	(5,10)	7.1009e-00(1.30e-01)+	7.2891e-00(6.04e-03)+	6.1310e-00(1.01e-01)+	6.9394e-00(6.72e-02)+	7.3266e-00(7.88e-03)
	(10,10)	7.1115e-00(8.53e-02)+	7.3454e-00(6.78e-03)=	7.0055e-00(4.02e-02)+	7.0487e-00(7.54e-02)+	7.3591e-00(2.82e-03)
	(20,10)	7.3583e-00(4.68e-02)=	7.3647e-00(2.46e-03)=	7.2520e-00(2.02e-02)+	7.3644e-00(5.84e-02)=	7.3685e-00(3.03e-03)
DF5	(10,5)	4.8072e-01(7.63e-03)+	4.9199e-01(7.93e-05)=	4.8976r-01(6.92e-04)+	4.8978e-01(5.97e-04)+	4.9258e-01(1.80e-04)
	(5,10)	4.8381e-01(4.04e-03)=	4.8382e-01(2.17e-03)=	3.4781e-01(1.81e-02)+	4.7041e-01(8.87e-03)+	4.8636e-01(1.46e-03)
	(10,10)	4.8318e-01(8.55e-03)=	4.9089e-01(7.70e-04)=	4.6138e-01(4.90e-03)+	4.4068e-01(1.02e-02)+	4.9176e-01(5.27e-04)
	(20,10)	4.9356e-01(1.44e-04)=	4.9340e-01(1.85e-04)=	4.8396e-01(2.18e-03)=	4.9328e-01(3.43e-03)=	4.9396e-01(1.74e-04)
DF6	(10,5)	7.3562e-01(6.78e-03)+	1.3395e-00(2.97e-01)+	8.8397e-01(9.43e-02)+	1.3164e-00(5.78e-02)+	1.6510e-00(1.26e-01)
	(5,10)	7.2844e-01(8.07e-03)+	1.4921e-00(9.15e-02)=	5.4955e-01(9.03e-02)+	8.0532e-01(3.03e-02)+	1.4970e-00(7.83e-02)
	(10,10)	7.4772e-01(6.88e-03)+	1.5095e-00(7.42e-02)+	7.0653e-01(9.24e-02)+	8.8042e-01(5.53e-02)+	1.8148e-00(6.23e-02)
	(20,10)	7.9592e-01(4.52e-03)+	1.5495e-00(6.90e-02)+	7.1831e-01(9.81e-02)+	1.1654e-00(4.16e-02)+	1.8945e-00(1.04e-01)
DF7	(10,5)	1.0203e+01(4.96e-03)=	1.0250e+01(2.72e-02)=	8.3566e-00(9.29e-02)+	8.6803e-00(1.27e-01)+	1.0253e+01(6.97e-01)
	(5,10)	1.0298e+01(9.20e-02)-	1.0573e+01(4.12e-02)=	9.4064e-00(9.55e-02)+	9.0245e-00(1.75e-01)+	1.0585e+01(1.02e-01)
	(10,10)	1.0389e+01(8.33e-02)+	1.1068e+01(9.02e-02)=	9.5807e-00(1.20e-01)+	9.9691e-00(3.88e-02)+	1.1214e+01(8.72e-02)
	(20,10)	1.0697e+01(6.81e-02)+	1.1219e+01(1.59e-01)=	1.0571e-01(8.38e-02)+	1.0198e+01(7.70e-02)+	1.2119e+01(1.38e-02)
DF8	(10,5)	3.7895e-01(3.65e-02)+	5.4775e-01(5.23e-05)=	5.4400e-01(4.08e-03)+	5.4699e-01(4.36e-04)=	5.4799e-01(4.46e-05)
	(5,10)	5.3716e-01(1.35e-03)+	5.4388e-01(2.49e-04)=	4.9365e-01(1.35e-02)+	5.4256e-01(1.20e-03)+	5.4501e-01(1.14e-04)
	(10,10)	5.3880e-01(7.70e-04)+	5.4558e-01(7.37e-05)=	5.2750e-01(7.36e-03)+	5.4108e-01(1.16e-03)+	5.4608e-01(7.38e-05)
	(20,10)	5.4579e-01(4.58e-05)=	5.4664e-01(5.06e-05)=	5.3790e-01(8.47e-03)+	5.4573e-01(5.50e-04)=	5.4666e-01(3.96e-05)
DF9	(10,5)	3.7852e-01(2.65e-02)+	4.1196e-01(5.30e-03)+	3.5048e-01(1.12e-02)+	3.6914e-01(1.42e-02)+	4.3347e-02(1.06e-02)
	(5,10)	3.7584e-01(7.09e-03)+	3.9546e-01(7.62e-03)=	2.2901e-01(1.89e-02)+	3.1963e-01(2.82e-02)+	3.9618e-01(3.13e-02)
	(10,10)	3.8013e-01(2.23e-02)+	4.4341e-01(8.85e-03)=	3.4486e-01(2.59e-01)+	2.4532e-01(1.47e-02)+	4.4359e-01(1.47e-02)
	(20,10)	4.5322e-01(3.58e-02)=	4.5138e-01(4.30e-03)=	3.5493e-01(1.93e-02)+	4.4673e-01(2.28e-02)+	4.5369e-01(1.09e-02)
DF10	(10,5)	5.2031e-01(5.51e-04)+	5.2423e-01(9.88e-04)+	4.9223e-01(3.21e-02)+	5.2872e-01(4.12e-03)+	5.5765e-01(1.26e-03)
	(5,10)	5.1149e-01(1.40e-02)+	5.2062e-01(3.70e-03)+	3.5559e-01(2.85e-02)+	5.3543e-01(5.11e-03)+	5.5277e-01(1.30e-03)
	(10,10)	5.3226e-01(7.79e-03)+	5.3614e-01(2.03e-03)+	4.6785e-01(3.35e-02)+	5.4969e-01(4.02e-03)+	5.5968e-01(1.22e-03)
	(20,10)	5.4841e-01(7.44e-03)+	5.5038e-01(1.24e-03)+	4.7284e-01(4.19e-02)+	5.6398e-01(2.01e-03)+	5.6556e-01(1.23e-03)
DF11	(10,5)	5.7275e-02(1.65e-03)+	5.7826e-02(3.96e-05)=	5.7118e-02(2.96e-03)+	5.7595e-02(6.74e-04)=	5.7956e-02(6.51e-04)
	(5,10)	5.6174e-02(1.46e-04)+	5.7009e-02(2.44e-04)+	5.4534e-02(1.04e-03)+	5.6942e-02(1.43e-04)+	5.7394e-02(8.53e-05)
	(10,10)	5.7688e-02(8.67e-05)+	5.7958e-02(1.07e-04)=	5.7416e-02(2.92e-04)+	5.7835e-02(6.71e-04)=	5.8087e-02(5.76e-04)
	(20,10)	5.8357e-02(1.66e-03)=	5.8469e-02(6.62e-05)=	5.8238e-02(1.56e-04)+	5.8455e-02(1.64e-04)=	5.8573e-02(6.29e-05)
DF12	(10,5)	2.9456e-01(1.72e-03)+	3.0867e-01(5.42e-03)+	2.6582e-01(6.71e-03)+	3.7596e-01(3.76e-03)+	4.2578e-01(2.65e-03)
	(5,10)	2.7478e-01(3.41e-03)+	3.3765e-01(4.72e-03)+	2.4483e-01(6.97e-03)+	3.7205e-01(4.38e-03)+	4.1782e-01(2.58e-03)
	(10,10)	2.9941e-01(1.68e-03)+	3.6750e-01(5.52e-03)+	2.7867e-01(6.05e-03)+	3.7754e-01(3.49e-03)+	4.2746e-01(2.22e-03)
	(20,10)	3.4925e-01(1.10e-03)+	4.1048e-01(4.36e-03)+	3.0259e-01(6.92e-03)+	3.9851e-01(3.84e-03)+	4.3531e-01(1.43e-03)
DF13	(10,5)	2.4465e-00(3.75e-03)=	2.4463e-00(1.25e-03)=	2.4463e-00(2.62e-03)=	2.4402e-00(3.56e-02)=	2.4495e-00(1.26e-03)
	(5,10)	2.4381e-00(1.29e-03)=	2.4281e-00(1.35e-03)=	2.4165e-00(1.89e-03)=	2.4084e-00(1.96e-03)+	2.4378e-00(9.10e-04)
	(10,10)	2.4571e-00(4.63e-04)=	2.4527e-00(5.53e-04)=	2.4561e-00(2.01e-03)=	2.4495e-00(4.75e-02)=	2.4542e-00(1.21e-03)
	(20,10)	2.4664e-00(7.88e-04)=	2.4666e-00(1.82e-04)=	2.4626e-00(5.45e-04)=	2.4605e-00(7.58e-04)=	2.4648e-00(2.10e-04)
DF14	(10,5)	3.1668e-00(5.63e-04)+	3.1596e-00(7.94e-05)+	3.1794e-00(1.56e-02)+	3.1852e-00(2.35e-02)=	3.1854e-00(3.65e-04)
	(5,10)	3.1840e-00(4.68e-04)=	3.1750e-00(8.22e-04)+	2.9793e-00(1.63e-02)+	3.1768e-00(1.26e-02)+	3.1854e-00(6.33e-04)
	(10,10)	3.1870e-00(5.30e-04)=	3.1873e-00(4.85e-04)=	3.1863e-00(1.48e-02)=	3.1862e-00(2.25e-03)=	3.1868e-00(3.34e-04)
	(20,10)	3.1879e-00(6.56e-04)=	3.1881e-00(3.81e-04)=	3.1867e-00(7.62e-04)=	3.1876e-00(2.87e-04)=	3.1880e-00(3.07e-04)
1st/2nd/all		7/10/56	9/28/56	0/2/56	0/8/56	40/8/56

Based on the experimental results of different combinations of (τ_t, n_t), it can be seen that for each problem, as τ_t increases, the testing algorithm can obtain more evaluations, which leads to a gradual increase in the MIGD and MHV metrics. In addition, as n_t increases, indicating smaller severity of environmental changes in dynamic problems, the MIGD and MHV metrics also tend to improve gradually.

In addition, the statistics of the Wilcoxon rank sum test for all the testing problems are given in Table III. From the result, it can be seen that the proposed STT-MOEA/D algorithm is significantly better than the comparison algorithm for most of the testing problems. To further evaluate the performance of all algorithms on DF benchmark problems, the average performance levels on MIGD and MHV by Friedman's test [58] is shown in Table IV. From the comparison results,

TABLE III
WILCOXON RANK-SUM TEST ANALYSIS OF STT-DMOEA/D WITH FOUR STATE-OF-THE-ART COMPARISON ALGORITHMS IN TERMS OF MEAN AND STANDARD DEVIATION OF MIGD AND MHV METRICS

STT-DMOEA/D	$v.s.$	Tr-DMOEA/D	IGP-DMOEA/D	ISVM-DMOEA/D	PPS-DMOEA/D
MIGD	+	43/56	33/56	49/56	46/56
	=	3/56	12/56	0/56	5/56
	-	10/56	11/56	7/56	5/56
MHV	+	32/56	17/56	47/56	42/56
	=	19/56	34/56	8/56	13/56
	-	5/56	5/56	1/56	1/56

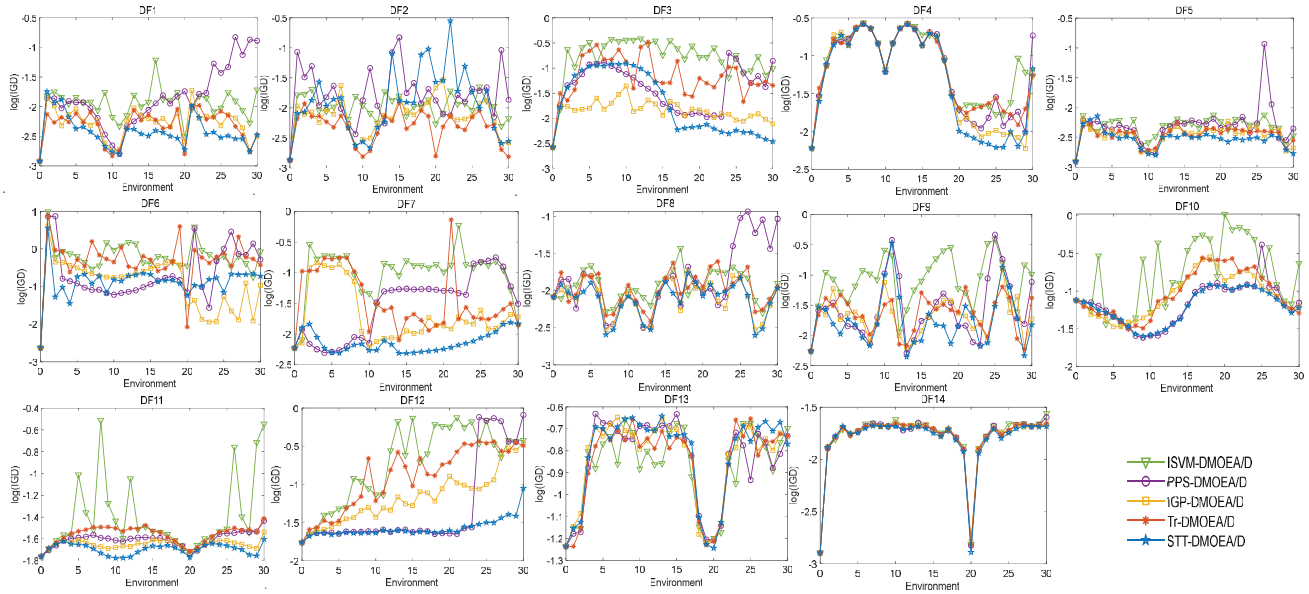


Fig. 9. Average IGD values versus environmental changes for five algorithms for different problems with $n_t = 10$ and $\tau_t = 10$.

TABLE IV
RANKINGS OF ALL COMPARISON ALGORITHMS BY THE FRIEDMAN TEST

Algorithms	MIGD	MHV
Tr-DMOEA/D	3.0714	3.0952
IGP-DMOEA/D	2.1429	2.1429
ISVM-DMOEA/D	4.4762	4.7619
PPS-DMOEA/D	3.5238	3.6190
STT-DMOEA/D	1.7857	1.3810

it is clear that for both MIGD and MHV metrics, STT-DMOEA/D ranks the best among all the rival algorithms.

2) Real-World Problem: The mean values of the MIGD metric obtained by the proposed STT-DMOEA/D and the four rival algorithms for the dynamic multiobjective operation optimization problem of continuous annealing production process (denoted as DMOOP-CAP) are given in Table V. From this table, it is clear that the proposed STT-DMOEA/D algorithm can succeed in achieving the best results for all three problems, each of which consists of two instances with different frequencies of environmental changes. Therefore, the experimental results on the real-world problem still illustrate the effectiveness of our proposed STT-DMOEA/D.

The proposed STT-DMOEA/D has been implemented in an operation optimization system for the continuous annealing of a major iron and steel enterprise in China, whose user interface is provided in Fig. 2 in the supplementary file. This system first reads in strip data and process information from the manufacturing execution system, then the static operation optimization will be performed to achieve the initial setting of control variables (operators can select a nondominated solution from the bottom-left section of the system). During production, an online monitoring module will check whether the composition and process constraints of the next strip or the production environment parameters have significantly changed. If so, the proposed STT-DMOEA/D algorithm will be performed to get the new nondominated solutions and the system can automatically select a solution according to the preference setting given by the operators.

F. Effectiveness Analysis of Adaptive Selection Strategy of Parameter d in D-BHT-ARIMA

As described in Section III-B, the adaptive selection strategy of parameter d is proposed in D-BHT-ARIMA based on the stability evaluation of $\mathcal{X}_{n::}$ for the nth subproblem (weight vector) in the objective space. In this section, further experiment was carried out to verify its effectiveness based on DF1-DF9. In this experiment, the adaptive selection strategy was compared with the traditional fixed strategy, i.e., parameter d was set to 0,1 and 2, respectively. The mean values of MIGD metric obtained by the four different settings (three fixed values and the adaptive one) are presented in Table II in

TABLE V
AVERAGE OF IGD OF STT-DMOEA/D AND FOUR COMPARISON ALGORITHMS IN THE CONTINUOUS ANNEALING PROBLEM

Problems	$(tau_t, 1/n_t)$	Tr-DMOEA/D	IGP-DMOEA/D	ISVM-DMOEA/D	PPS-DMOEA/D	STT-DMOEA/D
DMOOP-CAP1	(1,10)	0.2988	0.2893	0.2935	0.3651	0.2693
	(1,30)	0.2546	0.2697	0.2863	0.3538	0.2406
DMOOP-CAP2	(1,10)	0.2568	0.2275	0.2245	0.2495	0.2081
	(1,30)	0.2345	0.2139	0.2065	0.2191	0.1845
DMOOP-CAP3	(1,10)	0.2075	0.2185	0.2790	0.2374	0.2098
	(1,30)	0.1999	0.1923	0.2632	0.2175	0.1908

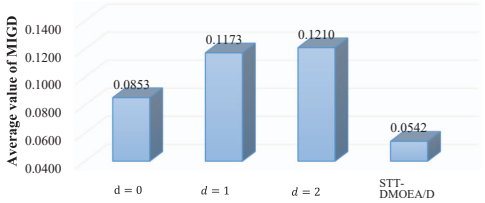


Fig. 10. MIGD obtained by different values of parameter d.

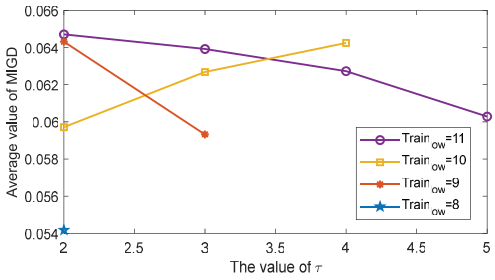


Fig. 11. Comparison of MIGD for different $\text{Train}_{\text{low}}$ and τ.

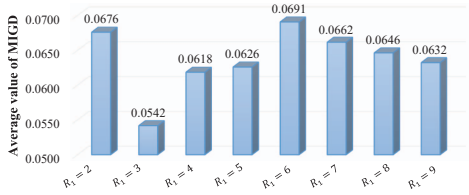


Fig. 12. MIGD obtained by different values of parameter R_1.

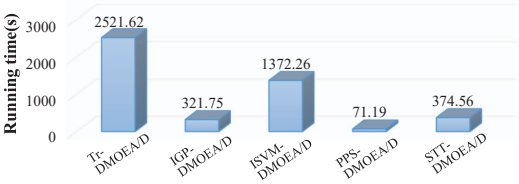


Fig. 13. Comparison of running time of different algorithms.

the supplementary file, from which it is clear that the proposed adaptive strategy of parameter d achieved the best results for all the problems except DF8. Based on the comparison result of the average value of all MIGDs shown in Fig. 10, it can be concluded that the proposed adaptive strategy is effective and helps to further improve the quality of the predicted new population.

G. Sensitivity Analysis of Parameters

In STT-DMOEA/D, there are two parameters that have significant impact on the performance of STT generator, i.e., the minimum number of training samples $\text{Train}_{\text{low}}$ and τ in the MDT operation. In the experiment, the value of $\text{Train}_{\text{low}}$ is selected from 8, 9, 10, 11 and all decomposable τ values are used accordingly. The mean values of MIGD metric for DF1-DF9 are given in Table III in the supplementary file, and visualized in Fig. 11. From the results, it appears that different settings of the two parameters really have some impact on the performance of STT-DMOEA/D, and the performance of STT-DMOEA/D tends to deteriorate as $\text{Train}_{\text{low}}$ increases. The main reason is that a large value of $\text{Train}_{\text{low}}$ means that more slices need to be copied to construct the training data in the data complementation strategy in the early environments, which in turn leads to inefficiency of the algorithm and causes inaccurate prediction. Therefore, based on the results, the parameter setting of $\text{Train}_{\text{low}} = 8$ and $\tau = 2$ is selected in the proposed STT-DMOEA/D.

In the Tucker decomposition, parameter R_1 is mainly used to define the size of the kernel tensor, which is the size of the matrix after extracting the information of the decision variables. Theoretically, the value of R_1 is not larger than the number of decision variables. In the experiment, R_1 is selected from 2 to 10 and the remaining parameters are fixed. The mean values of MIGD metric on DF1-DF9 are presented in Table IV in the supplementary file, and the average value of all MIGDs is shown in Fig. 12. From Fig. 12, it appears that the performance of STT-DMOEA/D first improves but gradually deteriorates as the value of R_1 increases. The main reason behind this phenomenon is that too small value of R_1 may result in too little extracted information, while too large value of R_1 may extract a lot of useless information and increase the computational effort. Therefore, based on the results, $R_1 = 3$ is selected in our STT-DMOEA/D.

H. Analysis of Running Time

The comparison of average running times of different algorithms on the personal computer with I7-12700 CPU is shown in Fig. 13. From this figure, it can be seen that both Tr-DMOEA/D and ISVM-DMOEA/D need more computational efforts than the other three algorithms. The main reason is that Tr-DMOEA/D and ISVM-DMOEA/D use a random search in the decision space for individuals that match to the requirements, and the randomness of the search significantly

increases the running time. In contrast, IGP-DMOEA/D, PPS-DMOEA/D and STT-DMOEA/D make direct predictions of new population, which substantially improves the efficiency of these algorithms. Depending on the complexity of the adopted prediction methods, there is a slight difference in the running times between IGP-DMOEA/D, PPS-DMOEA/D and STT-DMOEA/D. In summary, the efficiency of STT-DMOEA/D is competitive compared to other algorithms.

V. Conclusion

This article proposes a DMOEA based on spatial–temporal topological tensor (STT-DMOEA/D) for DMOPs. Unlike traditional DMOEAs, the proposed STT-DMOEA/D is based on a decomposition framework to mine the topological properties of the final populations in the objective space, and introduces the STT to represent the final populations, which can better characterize the spatial–temporal variation pattern of the final populations when environment changes. Based on the decomposition framework, the algorithm uses an improved tensor-based ARIMA model that works well for short TS to predict the new solution for each subproblem in the new environment, thus constructing a high-quality initial population. The proposed algorithm was tested on 14 dynamic benchmark problems covering different categories and the real-world dynamic multiobjective optimization problems in continuous annealing and compared with four state-of-the-art DMOEAs in the literature. The experimental results show that the proposed STT-DMOEA/D algorithm achieves the best results of MIGD and MHV metrics on most benchmark problems and on all real-world problems.

However, it is important to point out that the main limitation of the proposed STT prediction strategy is that it can only work well within the decomposition framework of MOEA/D, as it utilizes the relatively stable evolutionary trajectories of subproblems in the MOEA/D framework to predict new solutions. In addition, due to the limitation of MOEA/D, the proposed STT-DMOEA/D may not be able to solve some real-world dynamic multiobjective optimization problems with irregularly shaped PFs, such as long tail and sharp peak. Therefore, in the future research, in addition to the application of STT-DMOEA/D in the dynamic multiobjective operation optimization of practical production processes in the iron and steel industry, it is also necessary to investigate the self-adaptive adjustment of weight vectors and the corresponding stability evaluation of the evolution trajectory for the subproblem to guarantee the applicability of the proposed STT-DMOEA/D for more real-world problems.

References

[1] 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.

[2] J. Ding, C. Yang, Q. Xiao, T. Chai, and Y. Jin, "Dynamic evolutionary multiobjective optimization for raw ore allocation in mineral processing," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 1, pp. 36–48, Feb. 2019.

[3] M. Liu, "Robotic online path planning on point cloud," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1217–1228, May 2016.

[4] M. Mavrovouniotis, F. M. Muller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1743–1756, Jul. 2017.

[5] L. Tang and Y. Meng, "Data analytics and optimization for smart industry," *Front. Eng. Manage.*, vol. 8, no. 2, pp. 157–171, 2021.

[6] L. Tang, Z. Li, and J.-K. Hao, "Solving the single row facility layout problem by K-medoids memetic permutation group," *IEEE Trans. Evol. Comput.*, vol. 27, no. 2, pp. 251–265, Apr. 2023.

[7] C. Cruz, J. R. González, and D. A. Pelta, "Optimization in dynamic environments: A survey on problems, methods and measures," *Soft Comput.*, vol. 15, no. 7, pp. 1427–1448, 2011.

[8] S. Jiang, J. Zou, S. Yang, and X. Yao, "Evolutionary dynamic multiobjective optimisation: A survey," *ACM Comput. Surveys*, vol. 55, no. 4, pp. 1–47, 2022.

[9] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, and X. Yao, "A survey of evolutionary continuous dynamic optimization over two decades—Part A," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 609–629, Aug. 2021.

[10] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, and X. Yao, "A survey of evolutionary continuous dynamic optimization over two decades—part B," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 630–650, Aug. 2021.

[11] 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.

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

[13] S. Li, S. Yang, Y. Wang, W. Yue, and J. Qiao, "A modular neural network-based population prediction strategy for evolutionary dynamic multi-objective optimization," *Swarm Evol. Comput.*, vol. 62, Apr. 2021, Art. no. 100829.

[14] 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.

[15] 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 Trans. Evol. Comput.*, vol. 26, no. 4, pp. 690–703, Aug. 2022.

[16] L. Yan et al., "Manifold clustering-based prediction for dynamic multiobjective optimization," *Swarm Evol. Comput.*, vol. 77, Mar. 2023, Art. no. 101254.

[17] K. Yu et al., "A correlation-guided layered prediction approach for evolutionary dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 5, pp. 1398–1412, Oct. 2023.

[18] 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.

[19] L. Yan et al., "Interindividual correlation and dimension-based dual learning for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 6, pp. 1780–1793, Dec. 2023, doi: 10.1109/TEVC.2023.3235196.

[20] H. Zhang, J. Ding, M. Jiang, K. C. Tan, and T. Chai, "Inverse Gaussian process modeling for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 11240–11253, Oct. 2022.

[21] 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.

[22] 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.

[23] Y. Zhou, H. Lu, and Y.-M. Cheung, "Probabilistic rank-one tensor analysis with concurrent regularizations," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3496–3509, Jul. 2021.

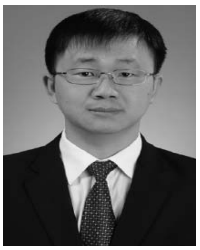
[24] S. B. Gee, K. C. Tan, and H. A. Abbass, "A benchmark test suite for dynamic evolutionary multiobjective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 461–472, Feb. 2017.

[25] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[26] K. Deb, N. Udaya Bhaskara 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. 4th Int. Conf. Evol. Multi-Criterion Optim.*, 2007, pp. 803–817.

[27] K. Zhang, C. Shen, X. Liu, and G. G. Yen, "Multiobjective evolution strategy for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 974–988, Oct. 2020.

[28] X. Ma, J. Yang, H. Sun, Z. Hu, and L. Wei, "Multiregional co-evolutionary algorithm for dynamic multiobjective optimization," *Inf. Sci.*, vol. 545, pp. 1–24, Feb. 2021.
[29] 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.
[30] H. Xie, J. Zou, S. Yang, J. Zheng, J. Ou, and Y. Hu, "A decision variable classification-based cooperative coevolutionary algorithm for dynamic multiobjective optimization," *Inf. Sci.*, vol. 560, pp. 307–330, Jun. 2021.
[31] I. Hatzakis and D. Wallace, "Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach," in *Proc. 8th Annu. Conf. Genet. Evol. Comput.*, 2006, pp. 1201–1208.
[32] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi-criterion Optim.*, 2007, pp. 832–846.
[33] W. T. Koo, C. K. Goh, and K. C. Tan, "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment," *Memet. Comput.*, vol. 2, no. 2, pp. 87–110, 2010.
[34] 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.
[35] M. Rong, D. Gong, W. Pedrycz, and L. Wang, "A Multimodel prediction method for dynamic multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 290–304, Apr. 2020.
[36] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, "Multidirectional prediction approach for dynamic multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3362–3374, Sep. 2019.
[37] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, "A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 5099–5112, Dec. 2020.
[38] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, and C. A. Coello Coello, "Weighted pointwise prediction method for dynamic multiobjective optimization," *Inf. Sci.*, vol. 546, pp. 349–367, Feb. 2021.
[39] C. Wang, G. G. Yen, and M. Jiang, "A grey prediction-based evolutionary algorithm for dynamic multiobjective optimization," *Swarm Evol. Comput.*, vol. 56, Aug. 2020, Art. no. 100695.
[40] J. Zheng, Y. Zhou, J. Zou, S. Yang, J. Ou, and Y. Hu, "A prediction strategy based on decision variable analysis for dynamic multi-objective optimization," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100786.
[41] 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, Mar. 2019.
[42] X. F. Liu, Y. R. Zhou, and X. Yu, "Cooperative particle swarm optimization with reference-point-based prediction strategy for dynamic multiobjective optimization," *Appl. Soft Comput.*, vol. 87, Feb. 2020, Art. no. 105988.
[43] K. Yu et al., "A framework based on historical evolution learning for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, early access, Jun. 28, 2023, doi: 10.1109/TEVC.2023.3290485.
[44] J. Nie and K. Ye, "Hankel tensor decompositions and ranks," *SIAM J. Matrix Anal. Appl.*, vol. 40, no. 2, pp. 486–516, 2019.
[45] Q. Shi et al., "Block Hankel tensor ARIMA for multiple short time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 5758–5766.
[46] T. Yokota, B. Erem, S. Guler, S. K. Warfield, and H. Hontani, "Missing slice recovery for tensors using a low-rank model in embedded space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8251–8259.
[47] C. Liu, S. C. Hoi, P. Zhao, and J. Sun, "Online learning of ARIMA for time series prediction," in *Proc. AAAI*, 2016, pp. 1867–1873.
[48] M. Kallas, P. Honeine, C. Francis, and H. Amoud, "Kernel autoregressive models using Yule–Walker equations," *Signal Process.*, vol. 93, no. 11, pp. 3053–3061, 2013.
[49] R. Mushtaq. "Augmented Dickey–Fuller test." SSRN.com. 2011. [Online]. Available: https://dx.doi.org/10.2139/ssrn.1911068
[50] S. Jiang, S. Yang, X. Yao, K. Tan, M. Kaiser, and N. Krasnogor, "Benchmark problems for CEC2018 competition on dynamic multiobjective Optimisation," School Comput., Newcastle Univ., London, U.K., Rep. CEC2018, Dec. 2017. [Online]. Available: https://www.semanticscholar.org/paper/Benchmark-Problems-for-CEC2018-Competition-on-Jiang-Yang/9bb47fd3d6445d739b1e78aa2d177312d07fac1b
[51] 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.
[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] X. Wang, Z. Dong, L. Tang, and Q. Zhang, "Multiobjective multitask optimization-neighborhood as a bridge for knowledge transfer," *IEEE Trans. Evol. Comput.*, vol. 27, no. 1, pp. 155–169, Feb. 2023.
[55] 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.
[56] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.
[57] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.
[58] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.



Xianpeng Wang (Senior Member, IEEE) received the B.S. degree in materials and control engineering from Shenyang University, Shenyang, China, in 2002, and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, in 2007.

He is currently a Professor with the Key Laboratory of Data Analytics and Optimization for Smart Industry, Ministry of Education, and the Liaoning Engineering Laboratory of Operations Analytics and Optimization for Smart Industry, Northeastern University. He has published more than 50 papers in international journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, *European Journal of Operational Research*, *Applied Soft Computing*, *Computers and Operations Research*, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and *Information Sciences*. His research interests include multiobjective optimization, machine learning, production scheduling, modeling and optimization in process industries based on data analytics, decision support systems, and process operation optimization.

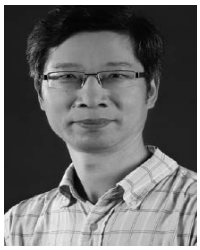
Yumeng Zhao received the B.S. degree in automation and the M.S. degree in control theory and control engineering with industrial engineering from the Lanzhou University of Technology, Lanzhou, China, in 2017 and 2020, respectively. She is currently pursuing the Ph.D. degree in control theory and control engineering with Northeastern University, Shenyang, China.

Her current research interests include dynamic multiobjective optimization and its applications.

Lixin Tang (Fellow, IEEE) received the B.Eng. degree in industrial automation, the M.Eng. degree in systems engineering, and the Ph.D. degree in control science and engineering from Northeastern University, Shenyang, China, in 1988, 1991, and 1996, respectively.

He is a member of Chinese Academy of Engineering, the Director of Key Laboratory of Data Analytics and Optimization for Smart Industry, Ministry of Education, China, and the Director and the Chair Professor of National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University. He has published many papers in international journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON POWER SYSTEMS, *Operations Research*, and *INFORMS Journal on Computing*. His research interests cover industrial intelligence and systems optimization theories and methods, covering data analytics and machine learning, deep learning and evolutionary learning, reinforcement learning and dynamic optimization, convex and sparse optimization, integer and combinatorial optimization, and computational intelligence-based optimization. Meanwhile, he applies the above theories and technologies to engineering applications in steel manufacturing industry, equipment/chip manufacturing industry, energy industry, logistics industry, and information industry.

Dr. Tang's paper published on *IISE Transactions* received the Best Applications Paper Award of 2017. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, *IISE Transactions*, *Journal of Scheduling*, and *International Journal of Production Research*. Meanwhile, he is on the Editorial Board of *Annals of Operations Research* and serves as an Area Editor for the *Asia–Pacific Journal of Operational Research*.



Xin Yao (Fellow, IEEE) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc. degree from the North China Institute of Computing Technologies, Beijing, China, in 1985, and the Ph.D. degree from USTC in 1990.

He is currently the Tong Tin Sun Chair Professor of Machine Learning with Lingnan University, Hong Kong, and a Part-Time Professor of Computer Science with the University of Birmingham, Birmingham, U.K. His research interests include evolutionary computation, ensemble learning, trustworthy artificial intelligence, and AI ethics.

Dr. Yao was a recipient of the 2001 IEEE DONALD G. FINK PRIZE PAPER AWARD for his Paper on Evolving Artificial Neural Networks, 2010, 2016, and 2017 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION OUTSTANDING PAPER AWARDS, the 2011 IEEE TRANSACTIONS ON NEURAL NETWORKS OUTSTANDING PAPER AWARD, the 2012 *Royal Society Wolfson Research Merit Award*, the 2013 IEEE CIS EVOLUTIONARY COMPUTATION PIONEER AWARD, the 2020 IEEE FRANK ROSENBLATT AWARD, and many other best paper awards at international conferences. He was the President of IEEE COMPUTATIONAL INTELLIGENCE SOCIETY (CIS) from 2014 to 2015 and the Editor-in-Chief from 2003 to 2008 of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He was a Distinguished Lecturer of IEEE CIS.