Knee Point-Based Imbalanced Transfer Learning for Dynamic Multiobjective Optimization

Min Jiang, *Senior Member, IEEE*, Zhenzhong Wang, Haokai Hong, and Gary G. Yen, *Fellow, IEEE*

***Abstract*—Dynamic multiobjective optimization problems (DMOPs) are optimization problems with multiple conflicting optimization objectives, and these objectives change over time. Transfer learning-based approaches have been proven to be promising; however, a slow solving speed is one of the main obstacles preventing such methods from solving real-world problems. One of the reasons for the slow running speed is that low-quality individuals occupy a large amount of computing resources, and these individuals may lead to negative transfer. Combining high-quality individuals, such as knee points, with transfer learning is a feasible solution to this problem. However, the problem with this idea is that the number of high-quality individuals is often very small, so it is difficult to acquire substantial improvements using conventional transfer learning methods. In this article, we propose a knee point-based transfer learning method, called KT-DMOEA, for solving DMOPs. In the proposed method, a trend prediction model (TPM) is developed for producing the estimated knee points. Then, an imbalance transfer learning method is proposed to generate a high-quality initial population by using these estimated knee points. The advantage of this approach is that the seamless integration of a small number of high-quality individuals and the imbalance transfer learning technique can greatly improve the computational efficiency while maintaining the quality of the solution. The experimental results and performance comparisons with some chosen state-of-the-art algorithms demonstrate that the proposed design is capable of significantly improving the performance of dynamic optimization.**

***Index Terms*—Domain adaptation, evolutionary dynamic multiobjective optimization, knee point, prediction, transfer learning.**

I. Introduction

Dynamic multiobjective optimization problems (DMOPs) refer to optimization problems involving multiple conflicting objectives that change over time. Many engineering problems can be formulated as DMOPs. For example, the motion planning of robots [1] requires path planning algorithms to continuously generate optimal actions according to the tasks at hand (i.e., minimizing energy consumption and maximizing stability) and the dynamic environments in which the robots operate. Another convincing example is industrial dynamic scheduling [2]; decision-makers should adjust their strategies to simultaneously achieve the competing goals of increasing productivity, decreasing costs and reducing energy consumption under constantly changing environments of fluctuating power demand, aging equipment, or faulty subsystems. Therefore, it is of great practical significance to solve DMOPs efficiently and effectively.

Various approaches have been proposed to solve DMOPs over the years [3]. For example, static multiobjective optimization algorithms (SMOAs) [4] can be directly used for DMOPs, and the methods simply reset the population when a change is detected. However, the population resetting approaches may inadvertently lead the search process in the wrong direction, so the efficiency of such algorithms is often very low. Multipopulation approaches [5]–[7], on the other hand, have been incorporated into evolutionary algorithms (EAs) to warrant sufficient population diversity. For example, Jin *et al.* [8] proposed a strategy based on reference points to partition the population into several subpopulations for enhancing population diversity. Diversity is indeed key to solving DMOPs, but methods that only use multiple populations to enhance diversity remain inefficient. For example, when large changes occur, generating multiple populations is still a major hurdle.

In recent years, prediction-based DMOPs algorithms have gained much attention. This class of methods predicts the state of the changing environment by exploiting different machine learning techniques and then makes a decision such that the algorithms can accommodate the changes in advance. However, most prediction-based methods assume that the "past" and "future" states are correlated, which may be valid for slowing or minor changing dynamics. When more drastic changes occur, the prediction models often perform poorly, which makes it difficult to achieve the original goal of prediction.

Prediction models based on transfer learning [9]–[11] have been considered to be promising. The advantage of this kind of approaches is that they can reuse the existing "experience" to obtain a high-quality initial population. For example, in [9] and [10], the domain adaptation method was introduced to predict the Pareto optimal front (POF) in the new environments, and the extensive experimental results confirm the quality of the solutions gained. However, the existing methods often require a long training time, which is a major

Manuscript received November 6, 2019; revised February 9, 2020 and May 1, 2020; accepted June 16, 2020. Date of publication June 22, 2020; date of current version January 29, 2021. This work was supported by the National Natural Science Foundation of China under Grant 61673328. *(Corresponding author: Gary G. Yen.)*

Min Jiang, Zhenzhong Wang, and Haokai Hong are with the School of Informatics, Xiamen University, Xiamen 361005, China.

Gary G. Yen is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: gyen@okstate.edu).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

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

Digital Object Identifier 10.1109/TEVC.2020.3004027

hurdle for certain DMOPs at hand. One of the reasons for the slow running speed is that the existing methods do not carefully select individuals during the environmental selection process, which not only results in a large amount of computing resources being wasted on low-quality individuals but also greatly increases the possibility of negative transfer [12]. Therefore, we believe that if we can transfer the knowledge possessed by those high-quality individuals (from the perspectives of convergence and diversity), we can build a more efficient and accurate prediction model for DMOP applications under various real-world complications.

In multiobjective optimization, knee points are a subset of nondominated solutions that are characterized by a slight improvement in one objective along with some serious degradations of at least one other objective. A remarkable advantage of knee points showed that these individuals possess higher hypervolume (HV) values according to [13]. Research [14] showed that whenever a set of solutions completely dominates another set of solutions, the HV of the former is greater than that of the latter. At the same time, the HV index can simultaneously measure the convergence and diversity of the solution set. These two properties which greatly facilitate the process of evolutionary progression are especially important for the design of dynamic SMOAs.

Therefore, we believe that a combination of knee points and transfer learning is an effective way to solve the problems of the existing methods. However, since the number of knee points is far lower than the number of other types of solutions, how to address the problem of imbalanced data becomes the key to realize this idea.

In this article, a knee-point transfer learning-based dynamic multiobjective EA, called KT-DMOEA, is proposed. The motivation of the proposed KT-DMOEA algorithm is that we believe for the purpose of multiobjective optimization the few high-quality solutions have the most essential knowledge to learn from, so only a few knee points need to be transferred to generate a better initial population.

The proposed method can be briefly summarized as follows. We divide the objective space into several subspaces, and each subspace owns its respective knee points; we call the knee points of subspaces, regional knee points. Once the change is detected, we feed these regional knee points into a probabilistic prediction model, named the trend prediction model (TPM), to obtain special individuals, called estimated knee points under the new environment. Subsequently, an imbalanced transfer learning-based method, so called knee point imbalanced transfer, is used to construct an initial population by exploiting the estimated knee points.

The contributions of this work are as follows: first, knee points are effective in both accelerating the convergence and maintaining the diversity of a population of individuals. Therefore, reuse of the knowledge of a few high-quality knee points rather than a large number of common individuals greatly improves the performance of SMOAs in solving DMOPs. To the best of our knowledge, no similar work has been reported on transferring knee points to enhance the performance in solving DMOPs. Second, imbalanced transfer learning is proposed to solve DMOPs. The experimental results show that the proposed algorithm can greatly improve the convergence speed and the quality of the solutions.

The remainder of this article is organized as follows: in Section II, we introduce some basic concepts of DMOPs and then discuss the existing works in this technical field. In Section III, we propose a knee point-based imbalanced transfer learning method, called KT-DMOEA, for solving dynamic multiobjective evolutionary optimization problems. In Section IV, we present the experimental results of the proposed approach on various benchmark functions and discuss the comparison with other state-of-the-art algorithms. In Section V, we draw a conclusion of this article, which naturally leads to an outline of the future research directions.

II. Preliminary Studies and Related Research

In this section, we will briefly describe the definition of a multiobjective optimization problem [15], and next we will discuss the characteristics of knee points and why these individuals are used for prediction and transfer. Finally, we will discuss the related works to motivate our proposed design.

A. DMOPs

The mathematical form of DMOPs is as follows:

$$\begin{aligned}\textbf{Minimize}\quad & F(x,t) = \langle f_1(x,t), f_2(x,t), \ldots, f_M(x,t)\rangle \\ \text{s.t.}\quad & x \in \Omega\end{aligned}$$

where $x = <x_1, x_2, \ldots, x_n>$ is the decision vector and t is the time or environment variable. $f_i(x,t) : \Omega \rightarrow \mathbb{R}$ $(i = 1, \ldots, M)$. $\Omega = [L_1, U_1] \times [L_2, U_2] \times \cdots \times [L_n, U_n]$. $L_i, U_i \in \mathbb{R}$ are the lower and upper bounds of the ith decision variable, respectively. The aim of solving DMOPs is to search for the set of solutions at different environments, so that all the objectives are as minimal as possible.

Definition 1 (Dynamic Decision Vector Domination): At time t, a decision vector x_1 Pareto-dominates another vector x_2 denoted by $x_1 \succ_t x_2$, if and only if

$$\begin{cases}\forall i = 1, \ldots, m, & f_i(x_1,t) \leq f_i(x_2,t)\\ \exists i = 1, \ldots, m, & f_i(x_1,t) < f_i(x_2,t)\end{cases} \tag{1}$$

Definition 2 (Dynamic Pareto-Optimal Set (DPOS): If a decision vector x^* at time t satisfies

$$\text{DPOS}_t = \left\{x^* | \nexists x, x \succ_t x^*\right\} \tag{2}$$

then x^* is called dynamic Pareto-optimal solution, and the set of dynamic Pareto-optimal solutions is called the DPOS.

Definition 3 [Dynamic Pareto-Optimal Front (DPOF)]: At time t, the DPOF is the corresponding objective vectors of the DPOS

$$\text{DPOF}_t = \left\{F\left(x^*, t\right) | x^* \in \text{DPOS}_t\right\}.$$

B. Knee Points

In multiobjective optimization, among a number of Pareto optimal solutions, knee points are a subset of nondominated solutions that are characterized by a slight improvement in one objective along with significant degradation of at least one other objective. From the perspective of multiple criteria

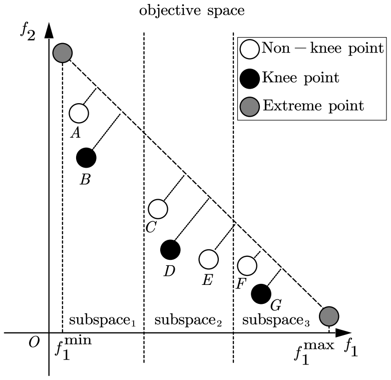

Fig. 1. For a bi-objective minimization problem, point *D* has the largest distance to the extreme line *L*, so it is chosen as the knee point. The objective space is divided into three subspaces, so there are three regional knee points *B*, *D*, and *G*.

decision making, if there is no other problem-specific or user-specific preferences, the choice of the knee point in the POF as a solution is naturally preferred [13]. Many studies [13], [16] have proven that the knee point is considered better than the other points on the POF for the metric of HV, and these results also show that the higher the HV is, the better the convergence and diversity of the population are. There are many methods to identify knee points, for example, a class of methods uses angles to determine whether a point is a knee point. A solution and the two neighbors of the solution can form two lines, and the two lines form an angle, so the angle is regarded as a criterion of whether a solution is a knee or not. If a solution has the largest angle among all of the angles, this solution is chosen as the knee point. Other methods are based on the expected marginal utility (EMU) [17], and the EMU measures the utility of a solution to each weight vector. For a solution that achieves the best performance along one weight vector, the EMU of this solution along this vector is defined as the additional cost after this solution is replaced by the second best solution. The solution with the largest marginal utility among all the weight vectors is considered the knee solution.

A simple way to find the knee point was proposed in [18], and this method determines whether a point is a knee point by calculating the distance from the point to an extreme line or a hyperplane. For a biobjective minimization problem, as shown in Fig. 1, the extreme line L is defined as follows:

$$L : a \cdot f_1 + b \cdot f_2 + c = 0 \tag{3}$$

where a, b, and c can be determined by two extreme points in the nondominated set. Thus, the distance from a solution $S(x_s, y_s)$ to the extreme line L can be calculated as

$$d(S, L) = \begin{cases} \frac{|ax_s+by_s+c|}{\sqrt{a^2+b^2}}, & ax_s + by_s + c < 0 \\ -\frac{|ax_s+by_s+c|}{\sqrt{a^2+b^2}}; & \text{otherwise.} \end{cases} \tag{4}$$

A knee point represents the solution that has the greatest distance to the extreme line L in the nondominated solution set. In Fig. 1, the point D is the greatest distance from the line L, so this point is considered the global knee point under the current population.

C. Related Works

In recent years, many advancements have been made in solving DMOPs, and most existing approaches can be divided into three main categories: 1) diversity maintenance methods; 2) memory-based methods; and 3) prediction-based methods.

The diversity maintenance methods tend to add variations to the solutions by using a certain technique when a change is detected. For example, in the SGEA method proposed in [19], the parameter η is used to control the portion of the population that is replaced with randomly created solutions. Every time a change occurs, the reinitialized population consists of 50% old solutions, η% randomly generated, and $(50 - \eta)$% prediction-guided solutions to improve the population diversity. DNSGA-II, proposed by Deb *et al.* [20], is also a method of maintaining diversity. There are two versions of the DNSGA-II, which are known as DNSGA-II-A and DNSGA-II-B. In the first version, the population is replaced by some solutions with randomly created individuals, while in the second version, the diversity is enhanced by replacing a percentage of the population with mutated solutions.

Jin *et al.* [8] proposed a strategy based on reference points. This method divides the population into several subpopulations according to the reference points. The centers of the same subpopulation are used to predict the centers of the same subpopulation under the new environment and increase the diversity of the solutions in the region. Yazdani *et al.* [21] proposed a cooperative coevolutionary multipopulation framework for solving large-scale dynamic optimization problems. The framework decomposes a large-scale dynamic optimization problem into a set of lower dimensional components and controls the budget assignment to components for tracking multiple moving optima. Zeng *et al.* [22] presented a general framework, called DCMOEA, for solving dynamic constrained optimization problems. DCMOEA handles the constraint difficulty by reducing the constraint boundary. Chen *et al.* [23] developed a dynamic two-archive EA (DTAEA) to handle DMOPs with a changing number of objectives. DTAEA simultaneously maintains two subpopulations. One subpopulation focuses on providing competitive selection pressure toward the POS, while the other subpopulation maintains a promising diversity.

A memory mechanism enables EAs to record past information, and memory-based methods use additional storage to implicitly or explicitly store useful information from previous generations to guide future searches. For example, Goh and Tan [24] proposed a coevolutionary multiobjective algorithm that hybridizes competitive and cooperative mechanisms to solve DMOPs. Each species subpopulation competes to represent a particular subcomponent of the multiobjective optimization problem, while the eventual winners cooperate to evolve for better solutions. Azzouz *et al.* [25] proposed an adaptive hybrid population management strategy using memory, local search, and random strategies to effectively handle environmental dynamism in DMOPs. The special feature of this algorithm is that it adjusts the number of memory and random solutions to be used according to the severity of the change.

Prediction-based methods [26], [27] for tracking moving optima have drawn increasing interest, and a variety of prediction-based methods have been proposed. Zhou *et al.* [28] proposed a population-based prediction strategy (PPS). In PPS, an autoregressive model was employed for predicting the population. A prediction model was built for predicting the POS center. Once the new POS center has been predicted, the POS manifold directly shifts to the new POS model. Rong *et al.* [29] presented a multidirectional prediction strategy to predict the location of the moving POS. In the method, a clustering algorithm was used to divide the population into different groups, where the number of clusters is adapted according to the intensity of the environmental change. Muruganantham *et al.* [30] proposed a predictive dynamic multiobjective EA (MOEA/D-KF) based on a Kalman filter. When a change in the environment was identified, the Kalman filter was used to guide the search toward a new POS through generating a new initial population

Peng *et al.* [31] put forward an exploration operator and exploitation operator to predict the new optimal solutions. In the prediction strategy, an optimal solution preservation mechanism is employed that reuses previously determined elite solutions. Ruan *et al.* [32] proposed a hybrid diversity maintenance method called the prediction and memory strategy (PMS) to improve prediction accuracy. Based on the direction of movement of the center points, the PMS uses the prediction model to relocate a number of solutions that are closed to the new POS. Gee *et al.* [33] presented the inverse modeling-based multiobjective algorithm (IM-MOEA/D) to solve DMOPs. IM-MOEA/D uses an inverse model to guide the search toward promising decision regions. Bu *et al.* [34] developed a dynamic species-based particle swarm optimization algorithm, and an ensemble of locating and tracking feasible region strategies was proposed to handle different types of dynamics in constraints. Gong *et al.* [35] provided a cooperative coevolutionary optimization framework for dynamic interval multiobjective optimization problems. A strategy divides the decision variables into two groups according to the interval similarity between each decision variable and interval parameters, and these two subpopulations were used to cooperatively optimize decision variables. Rong *et al.* [36] presented a multimodel prediction approach (MMP), and the method detected the type of change and then selected an appropriate prediction model to generate an initial population.

The methods based on transfer learning can be considered a branch of predictive methods. Jiang *et al.* [9] indicated that the solution distributions under different environments may not always be independent and identical (IID), so a framework, called Tr-DMOEA, based on transfer learning, was proposed to predict an effective initial population for solving DMOPs. The basic idea of the Tr-DMOEA is to map the POF in the past environment into the latent space and then use these mapped solutions to construct a population that can be used to search for the POF under a new environment.

The transfer learning-based approach does appreciably improve the quality of the solutions, but one problem of this approach that needs to be addressed is how to speed up the search process while improving the quality of the solution. As mentioned above, the existing methods spend a large amount of computing resources on "mediocre individuals," and a large number of transfers increase the likelihood of negative transfer. Therefore, we believe that if we can transfer the knowledge possessed by "high-quality" individuals, we can obtain a more efficient DMOP algorithm.

III. Proposed Method

A. Regional Knee Points

The proposed algorithm divides the objective space into several subspaces, and each subspace has one respective knee point, the regional knee point. Then, we feed regional knee points into TPM to obtain the estimated knee points under the new environment so that these estimated knee points can be used as samples of the target domain to assist transfer learning.

To obtain the regional knee points, we randomly select the mth objective function and divide the whole objective space evenly into p subspaces. Assuming that at time t, the maximum value of the mth function is $f_{m,t}^{\max}$, and the minimum value is $f_{m,t}^{\min}$, then the size of each subspace at time t is

$$\text{size}_{m,t,i} = \frac{f_{m,t}^{\max} - f_{m,t}^{\min}}{p}. \tag{5}$$

The lower bound of each subspace can be calculated as

$$LB_{m,t,i} = f_{m,t}^{\min} + (i-1) \times \text{size}_{m,t} \tag{6}$$

while the upper bound is calculated according to the following formula:

$$UB_{m,t,i} = f_{m,t}^{\min} + i \times \text{size}_{m,t} \tag{7}$$

where $i = 1, \ldots, p$.

Once we have identified each subspace, the regional knee point of this subspace can be determined. In this research, a regional knee point can be identified by the mentioned methods in Section II. Simply, we use the Das method [18] to find the point farthest from the extreme line, and Fig. 1 presents a simple example of this approach. In this example, the objective space is divided into three subspaces. Points B, D, and G are the farthest points in the three subspaces from the extreme line L, respectively, so these three points are regarded as the regional knee points.

B. Estimated Knee Points

In our approach, the estimated knee points are obtained by TPM based on these regional knee points, and Fig. 2 shows a schematic of the TPM method.

TPM consists of four steps. First, [as shown in Fig. 2(a)], for a subspace i, we can obtain a vector, $\vec{v}_i = k_i^{t-1} - k_i^{t-2}$, based on the regional knee points k_i^{t-2} at time $t-2$ and k_i^{t-1} at time $t-1$. This vector represents the trend of motion of the knee point in this region.

Second, for an $n-$dim motion vector $\vec{v}$, we express the vector in the polar coordinate $\vec{v} =< \varphi^1, \ldots, \varphi^{n-1}, r >$, where φ^j $(1 \leq j \leq n-1)$ are the angular coordinates and r is the radial coordinate [Fig. 2(b)]. To increase the speed of the calculation, we assume that the length of the vector is always the same,

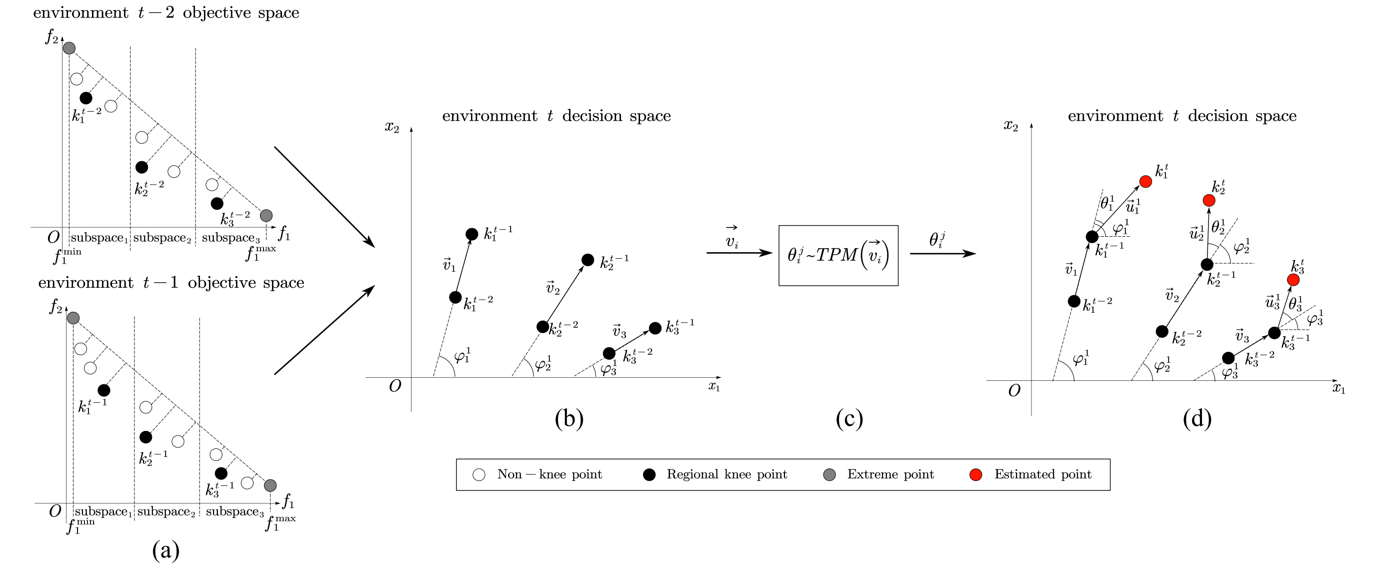

Fig. 2. Schematic of the TPM method. (a) Top and the bottom subfigures are objective spaces at environments $t-2$ and $t-1$, respectively. Each of the objective space is divided into three subspaces. For each subspace, a regional knee point is identified. For example, k_1^{t-1} represents the regional knee point in subspace$_1$ at time $t-1$. (b) Motion vectors, $\vec{v}_i = k_i^{t-1} - k_i^{t-2}$, are calculated according to the region knee points over two consecutive environments. (c) For a given motion vector $\vec{v}_i$, sample deflected angles θ_i^j by using the TPM. (d) Generate estimated knee points by using (10).

which implies the radial coordinate r is equal to $|\vec{v}|$. The jth correlative angle of the vector $\vec{v}_i$ is calculated as

$$\varphi_i^j = \arctan\left(\frac{\sqrt{\sum_{d=j+1}^{n}\left(\vec{v}_i^d\right)^2}}{\vec{v}_i^j}\right) \tag{8}$$

where $\vec{v}_i^j$ represents the jth component of the vector $\vec{v}_i$, and n represents the dimension of the decision variable.

In the third step, for a given motion vector $\vec{v}_i$, the proposed algorithm generates multiple deflection angles for each angle coordinate and then chooses the deflection angle with the greatest probability based on the following probability density function:

$$\theta_i^j \sim \text{TPM}(\vec{v}_i) = \frac{e^{-\text{sign}\left(\theta_i^j\right)\cdot\frac{\theta_i^j}{|\vec{v}_i|}}}{\int_{-\pi}^{0} e^{\frac{\theta_i^j}{|\vec{v}_i|}}\,d\theta_i^j + \int_{0}^{\pi} e^{-\frac{\theta_i^j}{|\vec{v}_i|}}\,d\theta_i^j} \tag{9}$$

where θ_i^j is a deflection angle and $\vec{v}_i$ is the motion vector. We believe that for a certain DMOP, knee point will move in a certain direction with a high probability. In this equation, the direction of the deflection angle of 0° is the direction in which the knee point is most likely to move. Therefore, deflection angles closer to 0° have higher probability values.

In the fourth step [Fig. 2(d)], when we obtain the deflection angles, the position of the estimated knee point of the subspace i at time t, k_i^t, can be calculated as follows:

$$k_i^t = k_i^{t-1} + \vec{u}_i \tag{10}$$

where k_i^{t-1} is the estimated knee point of the subspace i at time $t-1$.

Algorithm 1: TPM

```
Input: The DMOP $F_t(\cdot)$, two regional knee point sets $k^{t-2}$ and
       $k^{t-1}$, the number of subspaces $p$.
Output: Estimated Knee Points $K_{estimated}$
Set $n$ as the dimension of decision vector;
for $i = 1$ to $p$ do
    for $j = 1$ to $n-1$ do
        $\vec{v}_i = k_i^{t-1} - k_i^{t-2}$;
        Compute $\varphi_i^j$ according to Formula (8);
        Sample $\theta_i^j$ from $TPM(\vec{v}_i)$;
    end
    for $j = 1$ to $n$ do
        Calculate $\vec{u}_i^j$ according to Formula (11);
    end
end
Compute $K_{estimated}$ according to Formula (10) ;
return $K_{estimated}$
```

The vector $\vec{u}_i$ is obtained by the following formulas:

$$\vec{u}_i^j = \begin{cases} r\cos\left(\varphi_i^1 + \theta_i^1\right), & j = 1 \\ r\prod_{d=1}^{j-2}\sin\left(\varphi_i^d + \theta_i^d\right)\cos\left(\varphi_i^{j-1} + \theta_i^{j-1}\right), & 1 < j < n \\ r\prod_{d=1}^{j-1}\sin\left(\varphi_i^d + \theta_i^d\right), & j = n \end{cases} \tag{11}$$

where $\vec{u}_i^j$ means the jth component of the vector $\vec{u}_i$. φ_i^j and θ_i^j represent the jth corresponding correlative angle and deflection angle, respectively. n represents the dimension of the decision variable.

The details of TPM are shown in Algorithm 1.

C. Knee Point-Based Imbalanced Transfer Learning

The distribution of the knee points may vary greatly at different environments, so it is difficult to accurately predict the knee points at the next moment using only TPM. Therefore, we propose to use a transfer learning technique, so called TrAdaboost [37], to improve the prediction accuracy.

The basic idea of TrAdaboost is to use a boosting technique to filter out data in the source domain that are dissimilar with the data in the target domain, and this method uses a base classifier to obtain a set of weak classifiers. Each weak classifier maps samples X to a label $Y(X) \in \{0, 1\}$, where $X = X_{\text{source}} \cup X_{\text{target}}$. The outputs of these weak classifiers are combined into a strong classifier.

However, compared with other types of solutions, such as nondominated solutions and dominated solutions, the number of knee points is very low, so this problem is considered a typical imbalanced learning problem [38]. When the data are imbalanced, the small number of samples is likely to be misclassified by the weak classifiers, which leads to faster convergence of the weights in the source domain. As a result, in the next iteration of training, these minority solutions have little impact on the model training; thus, the recognition rate of the prediction model for small samples is very low. Thus, it is not suitable to use TrAdaboost directly to predict the knee points in the new environment.

We propose an imbalanced version of the TrAdaboost method to address the problem. The obtained POS and some randomly generated solutions of the last time step are used to form the source domain X_{so}, and the estimated knee points and some randomly generated solutions of the new environment are treated as the target domain X_{ta}. A classifier is iteratively employed to learn a sequence of weak classifiers, $h_1, \ldots, h_{N_{\max}}$, from $D = \{x, y(x)\}$, where $x \in X_{ta} \cup X_{so}$, and $y(\cdot)$ labels knee points as 1, and nonknee points as 0.

To prevent a fast decreasing of the weight values of samples from the minority class, we set larger initial weights for the knee points in the source domain from the start, and the weights of the samples in the source and target domains are set as follows:

$$w_1(x) = \begin{cases} \frac{1}{p}, & \text{if } x \in X_{so} \text{ and } y(x) = 1 \\ \frac{1}{n-p}, & \text{if } x \in X_{so} \text{ and } y(x) = 0 \\ \frac{1}{m}, & \text{if } x \in X_{ta} \end{cases} \tag{12}$$

where p is the number of subspaces and n and m are the numbers of samples in the source and target domain, respectively.

A factor σ is introduced to create rules for updating the weights of the samples, and σ can be expressed as

$$\sigma = \frac{c_k \cdot I_k + 0.001}{c_{nk} \cdot I_{nk} + 0.001} \tag{13}$$

where c_{nk} and c_k represent the costs of the misclassifications of a nonknee point and a knee point in the source domain, respectively, and I_k and I_{nk} are the numbers of misclassifications of knee points and nonknee points, respectively.

The basic idea of updating the weights by σ is that there are only a small number of knee points in the training set. When the data are imbalanced, the small number of samples is likely to be misclassified by the weak classifiers, which leads to faster convergence of the weights in the source domain. To address this issue, we introduce the σ factor. When the cost of misclassifying the minority class is greater than that of the majority class, the σ is set at greater than one, so that multiplying the σ factor can prevent fast decreasing of the weight values of samples from the minority class, and the performance of prediction is improved. In this research, we set c_{nk} equal to 1, and $c_k = ((|X_{so}| - p) \cdot c_{nk})/p$, where p is the number of subspaces.

The weights of the samples of the source and target domains can be updated according to the following rules:

$$w_{i+1}(x) = \begin{cases} w_i \sigma \beta^{|h_i(x) - y(x)|}, & x \in X_{so} \text{ and } y(x) = 1 \\ w_i \beta^{|h_i(x) - y(x)|}, & x \in X_{so} \text{ and } y(x) = 0 \\ w_i \beta_i^{-|h_i(x) - y(x)|}, & x \in X_{ta} \end{cases} \tag{14}$$

where

$$\beta = 1/\left(1 + \sqrt{2 \ln |X_{so}| / N_{\max}}\right) \tag{15}$$

and β_i is the error rate of the weak classifier h_i and $N_{\max}$ denotes the maximum number of iterations. β_i can be calculated as

$$\beta_i = \frac{\epsilon_i}{1 - \epsilon_i} \tag{16}$$

where ϵ_i is the weighted error in the target domain

$$\epsilon_i = \sum_{x \in X_{ta}} \frac{w_i(x) \cdot |h_i(x) - y(x)|}{\sum_{x \in X_{ta}} w_i(x)}. \tag{17}$$

Whenever we update the weights of the samples, we obtain a new weak classifier h_i. When the training is finished, we can combine the $N_{\max}$ weak classifiers to obtain a strong classifier $h(\cdot)$

$$h(\cdot) = \text{sign}\left(\sum_{i=1}^{N_{\max}} \ln(1/\beta_i) h_i(\cdot)\right). \tag{18}$$

$h(\cdot)$ can predict whether a solution is a knee point or not. We can randomly generate a large number of solutions and input the solutions into $h(\cdot)$ for classification. Algorithm 2 depicts this process in detail.

D. Overall Algorithm

Algorithm 3 depicts the main program of the proposed algorithm. At the first two environments, the population is initialized randomly, then the initial population is optimized by a population-based SMOA, and the respective POS is obtained. When the environment changes, the knee points of the last two environments are identified. Based on the motion trend of knee points from the last two environments, the knee points K_{estimaed} in the new environment are estimated by TPM. In line 13, the estimated knee points in the K_{estimaed} are labeled as 1, while some random dominated nonknee points P_t are labeled as 0, then K_{estimaed} and P_t are merged into the target domain training set X_{ta}. Similarly, in line 14, the random dominated nonknee points P_{t-1} and the knee points k_{t-1} in the last one environment are merged into the source domain training set X_{so}. Afterward, some knee points $K_{\text{predicted}}$ are predicted by the knee point imbalanced transfer technique. The predicted knee points $K_{\text{predicted}}$ can guide the search toward the true POF, and the population-based SMOA is employed

Algorithm 2: Knee Points Imbalanced Transfer (KPIT)

```
Input: The two data sets $X_{so}$ and $X_{ta}$, the number of
       subspaces $p$.
Output: Predicted knee points $K_{predicted}$
1  $D = \{x, y(x)\}$, where $x \in X_{ta} \cup X_{so}$;
2  Set the number of iterations $N_{max}$;
3  Set the cost of non-knee points $c_{nk}$ and knee points $c_k$ that
   satisfies $p \cdot c_k = (|X_{so}| - p) \cdot c_{nk}$;
4  Initialize the weight vector $w_1$ according to Formula (12) and
   normalize $w_1$;
5  for $i = 1$ to $N_{max}$ do
6      Train a weak classifier $h_i$ with $D$ and $w_i$;
7      $I_k$ is the number of misclassifications for knee points on
       $D_{source}$;
8      $I_{nk}$ is the number of misclassifications for non-knee points
       on $D_{source}$;
9      Calculate balance factor: $\sigma = \frac{c_k I_k + 0.001}{c_{nk} I_{nk} + 0.001}$;
10     Calculate the weighted error $\epsilon_i$:
       $\epsilon_i = \sum_{x \in X_{ta}} \frac{w_i(x) \cdot |h_i(x) - y(x)|}{\sum_{x \in X_{ta}} w_i(x)}$;
11     Calculate $\beta = \frac{\epsilon_i}{1 - \epsilon_i}$ and $\beta = \frac{1}{(1 + \sqrt{2 \ln |X_{so}| / N_{max}})}$;
12     Update the weight vector
       $w_{i+1}(x) = \begin{cases} w_i \sigma \beta^{|h_i(x) - y(x)|}, x \in X_{so} \textit{ and } y(x) = 1. \\ w_i \beta^{|h_i(x) - y(x)|}, x \in X_{so} \textit{ and } y(x) = 0. \\ w_i \beta_i^{-|h_i(x) - y(x)|}, x \in X_{ta}. \end{cases}$
13 end
14 Get a classifier $h(\cdot) = sign(\sum_{i=1}^{N_{max}} ln(1/\beta_i) h_i(\cdot))$;
15 Randomly generate large number of solutions $x_{test}$;
16 Select $x \in x_{test}$ which $h(x) = 1$ as $K_{predicted}$;
17 return $K_{predicted}$
```

Algorithm 3: KT-DMOEA

```
Input: The Dynamic Optimization Function $F_t(\cdot)$, a $SMOA$, the
       number of subspaces $p$.
Output: the POS of the $F_t(\cdot)$ at the different moments
1  Initialization;
2  while the environment has changed do
3      $t = t + 1$;
4      if $t == 1 || t == 2$ then
5          Initialize randomly the population $initPop$;
6          $POS_t = \textbf{SMOA}(F_t(\cdot), initPop)$;
7          Generate randomly dominated solutions $P_t$;
8      end
9      else
10         Identify regional knee points $k^{t-2}$ and $k^{t-1}$ in $POS_{t-2}$
           and $POS_{t-1}$, respectively;
11         $K_{estimated} = \textbf{TPM}(F_t(\cdot), k^{t-2}, k^{t-1}, p)$;
12         Generate randomly dominated solutions $P_t$;
13         $X_{ta} = K_{estimated} \cup P_t$;
14         $X_{so} = k_{t-1} \cup P_{t-1}$;
15         $K_{predicted} = \textbf{KPIT}(X_{so}, X_{ta}, p)$;
16         $POS_t = \textbf{SMOA}(F_t(\cdot), K_{predicted})$;
17     end
18     return $POS_t$;
19 end
```

to optimize $K_{\text{predicted}}$ in line 16. It should be noted that since the number of knee points is far less than the population size, other individuals in this population are generated by adding Gaussian noise into $K_{\text{predicted}}$.

How to detect and identify dynamic changes is a crucial part of solving DMOPs. However, in this article, our focus is placed solely on how prediction can quickly optimize DMOPs once the change has been detected.

E. Complexity Analysis

The complexity of the TPM module of the KT-DMOEA algorithm is $O(N^2M)$, where M is the number of objectives and N is the population size. The knee point transfer of KT-DMOEA follows the computational complexities of weak classifiers. If we use a support vector machine (SVM) as a weak classifier, applying TrAdaboost to obtain h requires $O(N^2n)$, where n is the dimension of decision variables. The overall complexity of KT-DMOEA proposed in this work is $O(N^2M) + O(N^2n) = O(N^2n)$.

IV. Experiments

A. Algorithms in Comparison and Test Problems

The proposed KT-DMOEA is compared against several popular algorithms, including MOEA/D-SVR [39], Tr-DMOEA [9], MOEA/D-KF [30], PPS [28], and the baseline algorithm MOEA/D [40], which is modified as RI-MOEA/D to adapt to dynamic change, i.e., 10% of population are randomly reinitialized when the environment changes.

For a fair comparison, the baseline algorithm in all compared algorithm is chosen to be MOEA/D [40], and these compared algorithms are denoted as KT-MOEA/D, SVR-MOEA/D, Tr-MOEA/D, KF-MOEA/D, and PPS-MOEA/D, respectively, and most of the parameters of these algorithms are set according to the original references.

All compared algorithms are evaluated based on 14 DMOPs selected from CEC 2018 DMO DF benchmark [41] suite. The dynamics of a DMOP is defined as $t = (1/n_t)\lfloor \tau/\tau_t \rfloor$, where τ, n_t, and τ_t refer to the maximum generation, the severity of change, and frequency of change, respectively. A low n_t implies a severely changing environment, whereas a lower τ_t denotes a faster changing environment. $\lfloor \cdot \rfloor$ is the flooring operator.

We also carry out experiments on other 14 benchmark DMOPs from three benchmark suties, FDA [42], dMOP [24], and F [28]. Due to the space limitation, the experimental results are placed in the supplemental material.

B. Performance Indicators and Settings

In this study, the following metrics are used to evaluate the performance of the compared algorithms from different perspectives.

1) Inverted generational distance (IGD) [43]: The definition of the IGD is as follows:

$$\text{IGD}\big(\text{POF}^*, \text{POF}\big) = \frac{1}{n} \sum_{p^* \in \text{POF}^*} \min_{p \in \text{POF}} \left\| p^* - p \right\|^2 \quad (19)$$

where POF* is the true POF of a given multiobjective optimization problem, *POF* is an approximation set of POF* obtained by a SMOA and n is the number of individuals in the POF*.

TABLE I
MEAN AND STANDARD DEVIATION VALUES OF MIGD METRIC OBTAINED BY COMPETING ALGORITHMS FOR DIFFERENT DYNAMIC TEST FUNCTIONS UNDER VARIOUS TEST SETTINGS, WHEN THE DIMENSION OF DECISION VARIABLES IS 10

Problems	τ_t, n_t	KT-MOEA/D	PPS-MOEA/D	SVR-MOEA/D	Tr-MOEA/D	KF-MOEA/D	RI-MOEA/D
DF1	10,10	**0.0855±6.29e-02**	0.1002±6.67e-02(+)	0.0920±7.72e-02(+)	0.1152±1.09e-01(+)	0.1594±8.61e-02(+)	0.1222±1.70e-01(+)
	10,5	0.1093±8.37e-02	0.1573±1.43e-01(+)	**0.0996±9.21e-02(-)**	0.1065±4.70e-02(=)	0.1859±1.35e-01(+)	0.1217±7.73e-02(+)
	5,10	**0.1323±8.96e-02**	0.1820±1.38e-01(+)	0.1412±9.07e-02(+)	0.1778±9.09e-02(+)	0.2019±9.91e-02(+)	0.1906±1.06e-01(+)
DF2	10,10	**0.0733±5.75e-02**	0.0758±7.61e-02(+)	0.0846±6.28e-02(+)	0.0796±4.25e-02(+)	0.1052±5.76e-02(+)	0.0898±5.03e-02(+)
	10,5	**0.0698±4.87e-02**	0.1194±9.53e-02(+)	0.0837±6.97e-02(+)	0.0851±4.13e-02(+)	0.1225±9.97e-02(+)	0.0851±4.88e-02(+)
	5,10	0.1240±5.82e-02	**0.1222±5.90e-02(-)**	0.1335±7.36e-02(+)	0.1517±7.88e-02(+)	0.1467±7.13e-02(+)	0.1446±7.92e-02(+)
DF3	10,10	0.3510±2.12e-01	0.4233±2.61e-01(+)	0.3934±2.19e-01(+)	**0.3239±1.44e-01(-)**	0.3663±1.64e-01(+)	0.4207±2.15e-01(+)
	10,5	0.3946±2.41e-01	0.4194±2.37e-01(+)	251917.3601±1.78e+06(+)	**0.3340±1.43e-01(-)**	0.3832±2.24e-01(-)	0.4065±1.85e-01(+)
	5,10	0.4053±2.04e-01	0.4766±2.83e-01(+)	0.4153±1.66e-01(+)	**0.4008±1.32e-01(-)**	0.3615±1.50e-01(-)	0.4666±2.08e-01(+)
DF4	10,10	0.9643±5.86e-01	**0.9567±5.86e-01(-)**	1.0871±6.54e-01(+)	1.1678±6.00e-01(+)	1.2995±8.17e-01(+)	1.1112±5.86e-01(+)
	10,5	**0.9741±5.90e-01**	1.0125±5.70e-01(+)	1.1133±7.08e-01(+)	1.1431±6.50e-01(+)	1.2019±6.09e-01(+)	1.1639±5.85e-01(+)
	5,10	1.0266±6.49e-01	**0.9962±5.71e-01(-)**	1.0488±6.32e-01(+)	1.3115±6.34e-01(+)	1.2892±7.78e-01(+)	1.2736±5.20e-01(+)
DF5	10,10	**1.2744±2.08e+00**	1.3059±2.04e+00(+)	1615.3660±5.92e+03(+)	1.5886±2.49e+00(+)	1.3032±2.04e+00(+)	1.5440±2.38e+00(+)
	10,5	**1.3107±2.15e+00**	1.3589±2.01e+00(+)	1427.2753±9.29e+03(+)	1.6223±2.46e+00(+)	1.3509±2.23e+00(+)	1.5720±2.46e+00(+)
	5,10	**1.2956±2.03e+00**	1.3641±2.25e+00(+)	52.0965±3.61e+02(+)	1.8847±2.75e+00(+)	1.3630±1.96e+00(+)	1.8507±2.70e+00(+)
DF6	10,10	2.3480±2.87e+00	4.0579±4.60e+00(+)	2.9385±4.38e+00(+)	**1.8157±2.66e+00(-)**	3.1488±3.41e+00(+)	2.7378±3.09e+00(+)
	10,5	1.9362±3.61e+00	2.9077±3.33e+00(+)	2.7888±4.77e+00(+)	3.1819±5.92e+00(+)	2.9899±3.28e+00(+)	**1.5022±9.67e-01(-)**
	5,10	**3.0427±3.25e+00**	5.4113±6.42e+00(+)	4.1532±4.82e+00(+)	3.8332±4.75e+00(+)	3.7650±4.81e+00(+)	4.8446±5.15e+00(+)
DF7	10,10	2.2345±2.32e+00	4.1744±5.18e+00(+)	2.7529±3.77e+00(+)	**2.0291±2.42e+00(-)**	4.0054±4.94e+00(+)	2.9849±3.85e+00(+)
	10,5	**2.0195±1.48e+00**	3.6013±5.32e+00(+)	2.6277±4.30e+00(+)	2.1043±2.12e+00(+)	3.3491±4.19e+00(+)	2.3055±2.34e+00(+)
	5,10	**3.1397±4.24e+00**	5.4884±6.69e+00(+)	3.3760±4.15e+00(+)	3.1823±4.22e+00(+)	3.9788±4.79e+00(+)	3.3120±3.08e+00(+)
DF8	10,10	**0.8274±4.58e-01**	1.0943±5.98e-01(+)	0.9774±5.21e-01(+)	0.9792±5.19e-01(+)	1.1486±5.44e-01(+)	0.9517±5.08e-01(+)
	10,5	**0.8749±4.29e-01**	1.0704±4.75e-01(+)	0.9810±5.19e-01(+)	0.9949±4.79e-01(+)	1.1095±5.46e-01(+)	0.9940±4.59e-01(+)
	5,10	**0.8212±4.69e-01**	1.0187±5.63e-01(+)	0.9079±4.97e-01(+)	0.9122±4.65e-01(+)	1.0174±4.98e-01(+)	0.8663±4.45e-01(+)
DF9	10,10	1.6579±1.69e+00	1.7546±1.75e+00(+)	551.8122±3.83e+03(+)	**1.6306±1.60e+00(-)**	1.6846±1.62e+00(+)	1.6587±1.62e+00(=)
	10,5	1.7776±1.77e+00	1.4683±1.41e+00(-)	128.9441±8.88e+02(+)	1.4271±1.48e+00(-)	1.4305±1.38e+00(-)	**1.4032±1.13e+00(-)**
	5,10	1.8248±1.83e+00	1.7708±1.85e+00(-)	2.6563±1.94+00(+)	1.8170±2.06e+00(-)	**1.6811±1.49e+00(-)**	1.7829±1.99e+00(-)
DF10	10,10	**0.1880±1.34e-01**	0.1891±8.50e-02(+)	64.9037±2.84e+02(+)	0.1883±7.86e-02(=)	0.2144±1.51e-01(+)	0.2036±8.49e-02(+)
	10,5	**0.1950±1.39e-01**	0.2515±1.37e-01(+)	11.1049±6.54e+01(+)	0.2167±7.34e-02(+)	0.2366±9.22e-02(+)	0.2108±8.63e-02(+)
	5,10	**0.2000±1.12e-01**	0.2439±1.42e-01(+)	10.8698±5.26e+01(+)	0.2196±1.02e-01(+)	0.2419±8.71e-02(+)	0.2210±1.28e-01(+)
DF11	10,10	**0.1387±2.31e-02**	0.1948±7.28e-02(+)	237.6269±4.57e+02(+)	0.1953±4.96e-02(+)	0.1851±3.45e-02(+)	0.1916±4.77e-02(+)
	10,5	**0.1395±2.27e-02**	0.2743±1.07e-01(+)	372.4522±8.32e+02(+)	0.2622±8.64e-02(+)	0.2624±7.94e-02(+)	0.2560±9.15e-02(+)
	5,10	**0.1725±3.92e-02**	0.2142±9.17e-02(+)	287.1599±6.83e+02(+)	0.2373±8.16e-02(+)	0.1975±4.46e-02(+)	0.2289±6.99e-02(+)
DF12	10,10	1.0082±2.45e-01	1.1771±1.13e-01(+)	252.6115±6.25e+02(+)	1.0084±2.53e-01(=)	0.9890±2.71e-01(-)	**0.9390±2.83e-01(-)**
	10,5	1.0520±2.14e-01	1.1891±3.31e-02(+)	288.4699±6.96e+02(+)	1.0046±2.47e-01(=)	**0.9121±3.19e-01(-)**	0.9286±2.84e-01(-)
	5,10	0.9962±2.68e-01	1.1847±5.55e-02(+)	336.9431±7.04e+02(+)	0.9777±2.76e-01(-)	0.9591±3.05e-01(-)	**0.9198±2.97e-01(-)**
DF13	10,10	**1.3474±1.87e+00**	1.3958±1.70e+00(+)	1.3785±1.77e+00(+)	1.5068±1.92e+00(+)	1.4413±1.80e+00(+)	1.4751±1.94e+00(+)
	10,5	**1.3327±1.81e+00**	1.4124±1.83e+00(+)	1.4661±1.20e+00(+)	1.5078±1.99+00(+)	1.4782±1.96e+00(+)	1.5023±1.98e+00(+)
	5,10	**1.4130±1.93e+00**	1.4235±1.84e+00(+)	1.5361±1.94e+00(+)	1.7077±2.21+00(+)	1.5552±2.02e+00(+)	1.7365±2.38e+00(+)
DF14	10,10	**0.8381±1.38e+00**	0.8657±1.31e+00(+)	4.1883±5.27e+00(+)	0.9316±1.42e+00(+)	0.9065±1.37e+00(+)	0.9239±1.42e+00(+)
	10,5	**0.8482±1.37e+00**	0.8735±1.30e+00(+)	4.4402±5.48e+00(+)	0.9383±1.42e+00(+)	0.9194±1.32e+00(+)	0.9500±1.41e+00(+)
	5,10	**0.8818±1.38e+00**	0.8864±1.35e+00(=)	3.6944±4.41e+00(+)	1.0686±1.58e+00(+)	0.9784±1.46e+00(+)	1.0668±1.60e+00(+)

The MIGD metric is a variant of IGD that is defined as the average of the IGD values over some time steps in a given run:

$$\mathrm{MIGD}\big(\mathrm{POF}^*, \mathrm{POF}\big) = \frac{1}{|T|}\sum_{t\in T}\mathrm{IGD}\big(\mathrm{POF}_t^*, \mathrm{POF}_t\big) \quad (20)$$

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

2) The HV [44] is used to quantify the convergence and distribution of solutions obtained by the algorithm. The definition of the HV is as follows:

$$\mathrm{HV}(\mathrm{POF}, \mathrm{ref}) = \Lambda\left(\bigcup_{p\in\mathrm{POF}}\left\{p'|\mathrm{ref} \succ p' \succ p\right\}\right) \quad (21)$$

where Λ is the Lebesgue measure and ref $\in R^M$ is the reference point for the computation of HV. MHV metric is a variant of HV that is defined as the average of the HV values over some time steps in a given run

$$\mathrm{MHV}(\mathrm{POF}, \mathrm{ref}) = \frac{1}{|T|}\sum_{t\in T} HV(\mathrm{POF}_t, \mathrm{ref}). \quad (22)$$

In the experiments, the dimension of decision variables is set to 10, and the population size is set to 100 for bi-objective optimization problems and 150 for tri-objective optimization problems. All the problems are run ten times independently. There are three pairs of dynamic settings in our experiments as commonly chosen in literature: ($n_t = 10, \tau_t = 10$), ($n_t = 10, \tau_t = 5$), and ($n_t = 5, \tau_t = 10$). The total number of generations τ is fixed to be $50 * \tau_t$, which ensures there are 50 changes in each run.

We set the parameters of KT-MOEA/D as follows: the number of subspaces p is set to 10, and we choose the first objective function to divide the subspace. The parameters in SVM are set by default [45].

C. Performance on DF Problems

The statistical results of MIGD and MHV can be found in Tables I and II. At the bottom of these two tables, "+," "=," and "−" indicate that the KT-MOEA/D is statistically significantly better, indifferent or worse than the compared algorithms, respectively, according to the Wilcoxon test [46] (both are at the 5% significance level).

TABLE II
MEAN AND STANDARD DEVIATION VALUES OF MHV METRIC OBTAINED BY COMPETING ALGORITHMS FOR DIFFERENT DYNAMIC TEST FUNCTIONS UNDER VARIOUS TEST SETTINGS, WHEN THE DIMENSION OF DECISION VARIABLES IS 10

Problems	τ_t, n_t	KT-MOEA/D	PPS-MOEA/D	SVR-MOEA/D	Tr-MOEA/D	KF-MOEA/D	RI-MOEA/D
DF1	10,10	0.5187±1.74e-01	0.3433±1.03e-01(+)	0.5175±1.36e-01(=)	0.5205±1.43e-01(=)	0.4645±1.36e-01(+)	**0.5292±1.40e-01(-)**
	10,5	**0.5671±1.72e-01**	0.3291±1.17e-01(+)	0.4944±1.43e-01(+)	0.4920±1.33e-01(+)	0.4521±1.52e-01(+)	0.4951±1.37e-01(+)
	5,10	**0.5142±2.01e-01**	0.3379±1.47e-01(+)	0.4182±1.25e-01(+)	0.4955±1.39e-01(+)	0.4180±1.43e-01(+)	0.5002±1.20e-01(+)
DF2	10,10	**0.6477±1.14e-01**	0.4782±1.10e-01(+)	0.6287±1.02e-01(+)	0.6328±7.08e-02(+)	0.5750±1.06e-01(+)	0.6401±8.25e-02(=)
	10,5	**0.7566±5.26e-02**	0.4842±1.21e-01(+)	0.6474±8.90e-02(+)	0.6497±8.19e-02(+)	0.5707±1.19e-01(+)	0.6209±7.96e-02(+)
	5,10	**0.6920±8.05e-02**	0.4516±1.33e-01(+)	0.6372±8.95e-02(+)	0.6152±8.52e-02(+)	0.5874±1.00e-01(+)	0.6249±8.05e-02(+)
DF3	10,10	0.3980±2.23e-01	0.2164±2.34e-01(+)	0.2434±1.83e-01(+)	**0.4820±1.85e-01(-)**	0.4407±2.19e-01(-)	0.4301±2.19e-01(-)
	10,5	0.3909±2.65e-01	0.2347±2.26e-01(+)	0.2261±1.81e-01(+)	0.4699±1.99e-01(-)	0.4102±2.09e-01(-)	**0.4743±2.08e-01(-)**
	5,10	0.3753±2.52e-01	0.2014±2.22e-01(+)	0.1816±1.66e-01(+)	**0.4779±2.08e-01(-)**	0.4347±1.83e-01(-)	0.4520±2.21e-01(-)
DF4	10,10	0.4350±2.21e-01	0.2212±1.89e-01(+)	0.3513±2.77e-01(+)	**0.4507±1.72e-01(-)**	0.2919±2.59e-01(+)	0.4450±2.07e-01(-)
	10,5	0.3883±2.34e-01	0.1696±1.74e-01(+)	0.3660±2.72e-01(+)	**0.4633±1.07e-01(-)**	0.2939±2.50e-01(+)	0.4572±1.88e-01(-)
	5,10	0.3805±2.37e-01	0.2379±2.46e-01(+)	0.3945±2.56e-01(-)	0.4197±2.21e-01(-)	0.2544±2.63e-01(+)	**0.4225±2.10e-01(-)**
DF5	10,10	0.6646±1.35e-01	0.4551±6.78e-02(+)	0.0495±1.39e-01(+)	0.6788±9.19e-02(-)	**0.6955±1.14e-01(-)**	0.6693±1.05e-01(=)
	10,5	**0.6798±1.22e-01**	0.4610±4.64e-02(+)	0.0488±1.38e-01(+)	0.6784±8.54e-02(=)	0.6653±9.79e-02(+)	0.6657±9.39e-02(+)
	5,10	0.6422±1.59e-01	0.4532±9.23e-02(+)	0.0393±9.91e-02(+)	0.6639±1.06e-01(-)	0.6799±1.15e-01(-)	**0.6807±9.00e-02(-)**
DF6	10,10	0.8426±2.19e-01	0.4502±2.51e-01(+)	0.1936±3.27e-01(+)	**0.9208±1.08e-01(-)**	0.8913±1.77e-01(-)	0.9145±1.14e-01(-)
	10,5	**0.8774±2.36e-01**	0.4360±2.71e-01(+)	0.1349±2.10e-01(+)	0.8720±1.75e-01(=)	0.8728±1.58e-01(=)	0.8447±5.01e-02(+)
	5,10	0.8436±2.53e-01	0.5439±2.76e-01(+)	0.0659±1.50e-01(+)	0.8698±1.40e-01(-)	0.8712±2.06e-01(-)	**0.8900±1.42e-01(-)**
DF7	10,10	0.8681±1.15e-01	0.3839±2.30e-01(+)	**0.9322±2.74e-01(-)**	0.9302±8.03e-02(-)	0.8772±2.03e-01(-)	0.9045±1.13e-01(-)
	10,5	0.8455±2.45e-01	0.3899±2.69e-01(+)	0.9007±2.68e-01(-)	0.9043±6.50e-02(-)	0.8785±1.81e-01(-)	**0.9120±1.35e-01(-)**
	5,10	0.8792±1.89e-01	0.6058±3.23e-01(+)	**0.9333±2.05e-01(-)**	0.9071±1.43e-01(-)	0.8481±2.52e-01(+)	0.8876±1.37e-01(-)
DF8	10,10	0.6827±2.36e-01	0.3222±2.11e-01(+)	**0.8347±2.47e-01(-)**	0.8378±1.25e-01(-)	0.7399±1.99e-01(-)	0.7837±1.18e-01(-)
	10,5	0.6512±2.25e-01	0.3976±2.02e-01(+)	**0.9716±6.86e-02(-)**	0.8272±1.39e-01(-)	0.7683±2.19e-01(-)	0.8085±1.16e-01(-)
	5,10	0.6682±2.70e-01	0.4079±2.36e-01(+)	**0.8853±1.78e-01(-)**	0.7863±1.63e-01(-)	0.7304±2.45e-01(-)	0.7986±1.26e-01(-)
DF9	10,10	0.4527±2.41e-01	0.2979±1.88e-01(+)	0.2826±6.04e-02(+)	0.5347±2.32e-01(-)	0.5134±2.38e-01(-)	**0.5469±2.31e-01(-)**
	10,5	0.4951±2.67e-01	0.2889±2.93e-01(+)	0.2813±7.00e-02(+)	**0.5534±2.20e-01(-)**	0.5258±2.06e-01(-)	0.5186±2.12e-01(-)
	5,10	**0.5452±2.23e-01**	0.2927±2.12e-01(+)	0.2738±9.89e-02(+)	0.5360±2.50e-01(+)	0.5124±2.21e-01(+)	0.5450±2.32e-01(=)
DF10	10,10	**0.9118±3.72e-01**	0.7068±1.18e-01(+)	0.8159±3.19e-01(+)	0.8750±1.37e-01(+)	0.8417±1.27e-01(+)	0.8917±1.26e-01(+)
	10,5	**0.8518±3.47e-01**	0.4813±2.84e-01(+)	0.6836±4.73e-01(+)	0.8440±1.99e-01(+)	0.8204±1.98e-01(+)	0.8405±1.81e-01(+)
	5,10	**0.8604±2.25e-01**	0.6103±2.14e-01(+)	0.7460±3.51e-01(+)	0.7594±2.07e-01(+)	0.8412±1.90e-01(+)	0.8463±2.07e-01(+)
DF11	10,10	**0.8255±4.49e-01**	0.1755±1.07e-01(+)	0.7252±3.85e-01(+)	0.2665±1.96e-01(+)	0.2554±1.18e-01(+)	0.2768±1.90e-01(+)
	10,5	**0.8010±3.33e-01**	0.1716±9.52e-02(+)	0.6944±3.81e-01(+)	0.2804±2.15e-01(+)	0.2741±1.80e-01(+)	0.2956±2.27e-01(+)
	5,10	**0.7563±2.33e-01**	0.1699±9.82e-02(+)	0.6967±3.80e-01(+)	0.2611±2.05e-01(+)	0.2494±2.62e-01(+)	0.2549±2.13e-01(+)
DF12	10,10	0.1727±3.06e-01	0.2175±3.59e-01(-)	0.1576±3.49e-01(+)	0.3949±3.96e-01(-)	0.4715±4.11e-01(-)	**0.6451±3.83e-01(-)**
	10,5	0.2397±3.60e-01	0.2146±3.48e-01(+)	0.1108±3.88e-01(+)	0.4345±4.27e-01(-)	**0.5256±4.07e-01(-)**	0.5199±4.13e-01(-)
	5,10	0.2219±3.32e-01	0.2191±3.01e-01(+)	0.1408±4.10e-01(+)	0.4377±3.97e-01(-)	0.4407±3.94e-01(-)	**0.6104±3.94e-01(-)**
DF13	10,10	**0.6727±1.21e-01**	0.3836±1.07e-01(+)	0.1158±1.22e-01(+)	0.6680±7.08e-02(+)	0.6710±1.06e-01(=)	0.6703±8.73e-02(=)
	10,5	**0.6926±1.14e-01**	0.3803±1.13e-01(+)	0.1209±1.28e-01(+)	0.6578±7.09e-02(+)	0.6750±7.50e-02(+)	0.6843±7.14e-02(+)
	5,10	**0.6675±9.75e-02**	0.3906±8.58e-02(+)	0.0884±1.27e-01(+)	0.6659±8.70e-02(=)	0.6595±8.57e-02(+)	0.6626±9.14e-02(=)
DF14	10,10	**0.6352±2.19e-01**	0.4563±1.90e-01(+)	0.0565±1.20e-01(+)	0.6284±1.90e-01(+)	0.5876±2.19e-01(+)	0.6108±1.73e-01(+)
	10,5	**0.6178±2.36e-01**	0.4423±2.02e-01(+)	0.0619±1.19e-01(+)	0.6149±1.64e-01(=)	0.6092±1.82e-01(+)	0.6098±1.80e-01(+)
	5,10	**0.6011±2.36e-01**	0.4251±2.10e-01(+)	0.0450±8.79e-02(+)	0.5955±1.76e-01(=)	0.6001±1.67e-01(=)	0.5962±2.01e-01(=)

The results in Table I shows that KT-MOEA/D performs significantly better in MIGD metric than the competing algorithms on DF5, DF8, DF10, DF11, DF13, and DF14 under all configurations. On DF1, DF2, DF4, DF6, and DF7, KT-MOEA/D falls slightly behind the corresponding best-performing algorithms in one or two of the dynamic test settings. However, on DF3, DF9, and DF12, KT-MOEA/D performs worse. Fig. 3 plots the IGD values obtained by different algorithms after each change. These curves show that the curves obtained by the proposed method are at the bottom in most cases, and the curves of the proposed method are smoother, which means that the method not only performs better but also is more stable. Because the predictor of MOEA/D-SVR is depended heavily on the quality of the past solutions [39], the IGD curves often rise suddenly, and the performance is not stable. Table II records the experimental results for the MHV values. Compared with the IGD metric, the HV index [14] can better quantify the convergence and diversity of the solutions obtained by the algorithms, and the experimental results show that the proposed algorithm has advantages over the other methods in most cases.

To see the performance of KT-MOEA/D on problems with high-dimensional decision variables, the number of decision variables is set to 30, and is tested on all compared algorithms on DF1-DF14. The experimental results are presented in the supplemental material. The results show that KT-MOEA/D is better than other algorithms on nearly half of the test cases. However, the performance is lower than that of 10-D decision variables, which is because the increase of decision variables brings challenges to the prediction of knee point movement by TPM.

To see whether the proposed prediction model could be integrated into other MOEAs, we also replace the baseline SMOA by RM-MEDA [43] and these compared algorithms are denoted as KT-RM-MEDA, SVR-RM-MEDA, Tr-RM-MEDA, KF-RM-MEDA, and PPS-RM-MEDA, respectively. The experimental results are presented at Table S-I in the supplementary material.

From the experimental results, It is obvious that KT-MOEA/D performs not very well on DF3, DF9, and DF12 problems. DF3 has a simple dynamic POF [41], and Tr-MOEA/D is suitable to solve DMOPs with a similar sequential POF in changing environment [9]. DF9 have disconnected

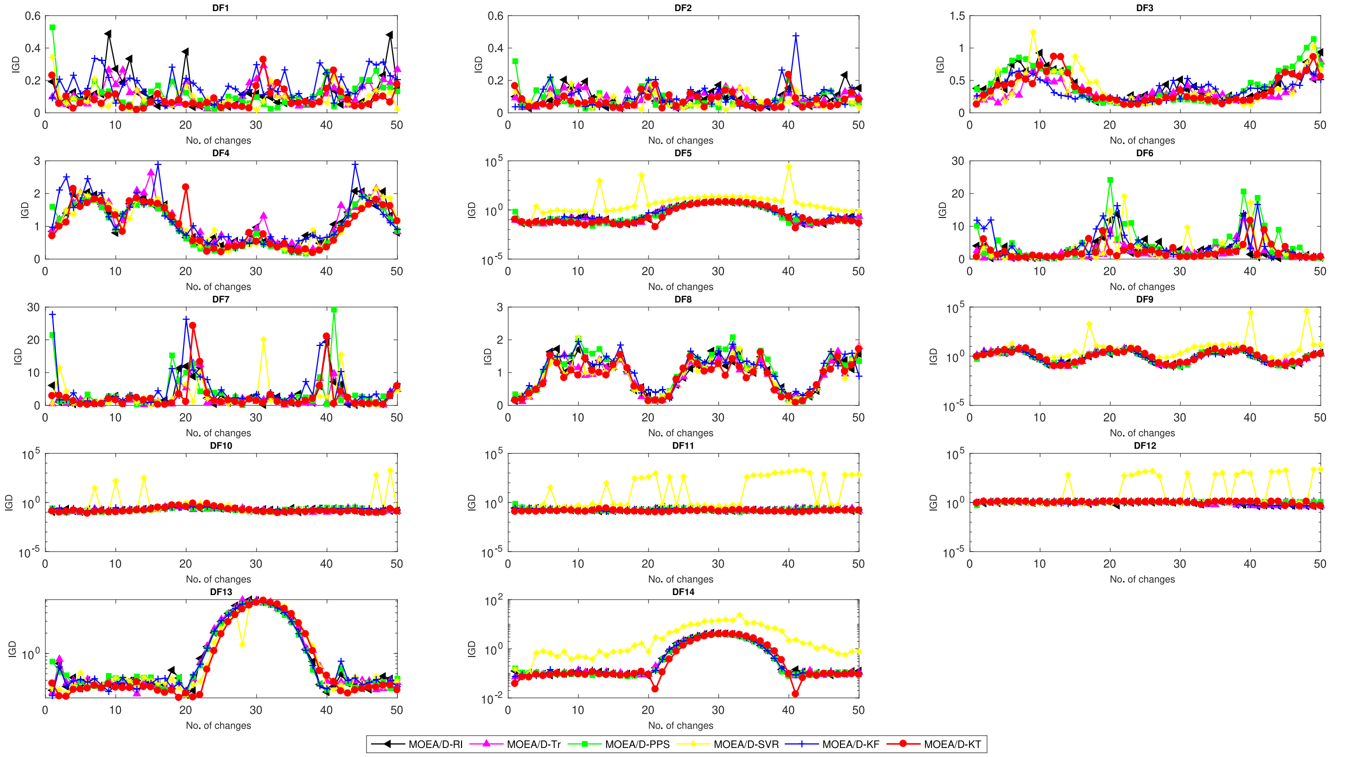

Fig. 3. Average IGD values at a configuration of $n_t = 10$ and $\tau_t = 10$, and the number of decision variables is 10.

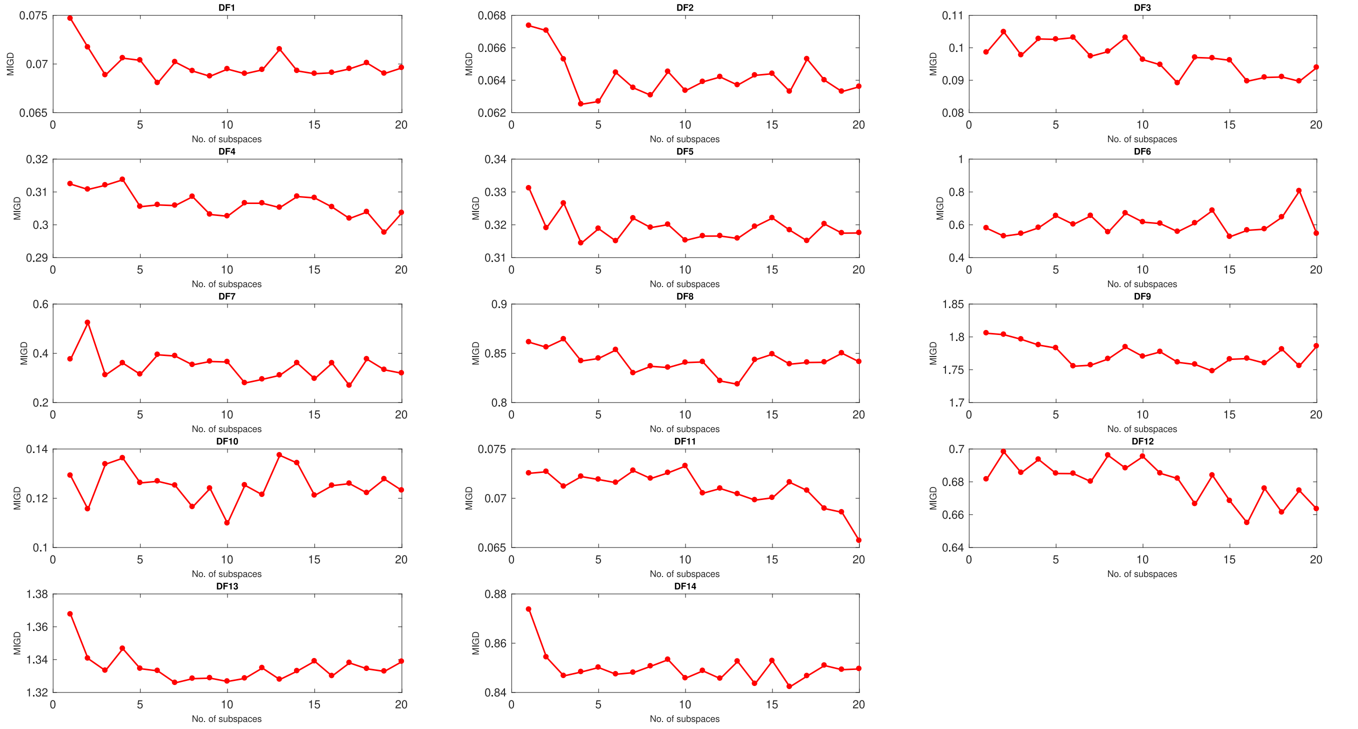

Fig. 4. Mean MIGD values obtained by KT-MOEA/D with different parameter p on different test functions.

POF segments, and DF12 have POF holes, both of them are considered complicated POFs. Therefore, the locations of knee points may change drastically, and it is difficult to estimate the position of knee point by TPM.

D. Influence of Parameter p in KT-MOEA/D

In the KT-MOEA/D, the parameter p plays a key role which decides the number of predicting knee points. The choice of parameter p affects prediction performance. In the

TABLE III
MEAN AND STANDARD DEVIATION VALUES OF MIGD METRIC OBTAINED BY KT-MOEA/D AND KT-MOEA/D$_\sigma$ UNDER DIFFERENT DIMENSION OF DECISION VARIABLES

Problems	τ_t, n_t	Decision Variables = 10		Decision Variables = 30	
		KT-MOEA/D	KT-MOEA/D$_\sigma$	KT-MOEA/D	KT-MOEA/D$_\sigma$
DF1	10,10	0.0855±6.29e-02	**0.0674±3.08e-02(-)**	**0.6950±4.61e-01**	0.7290±6.48e-01(+)
	10,5	0.1093±8.37e-02	**0.0900±7.96e-02(-)**	**0.9306±7.11e-01**	1.0057±7.56e-01(+)
	5,10	**0.1323±8.96e-02**	0.1342±9.16e-02(+)	0.9574±6.79e-01	**0.9477±6.57e-01(-)**
DF2	10,10	0.0733±5.75e-02	**0.0670±5.19e-02(-)**	0.4239±3.12e-01	**0.4047±2.55e-01(-)**
	10,5	**0.0698±4.87e-02**	0.0800±5.06e-02(+)	**0.5238±4.06e-01**	0.5739±4.30e-01(+)
	5,10	0.1240±5.82e-02	**0.1126±5.84e-02(-)**	0.6878±4.41e-01	**0.6574±4.79e-01(-)**
DF3	10,10	**0.3510±2.12e-01**	0.3663±1.53e-01(+)	0.9360±7.66e-01	**0.8844±5.38e-01(-)**
	10,5	0.3946±2.41e-01	**0.3595±2.17e-01(-)**	**0.9884±8.56e-01**	1.0635±9.65e-01(+)
	5,10	**0.4053±2.04e-01**	0.4309±2.95e-01(+)	1.0671±8.69e-01	**1.0443±7.86e-01(-)**
DF4	10,10	**0.9643±5.86e-01**	1.0112±6.50e-01(+)	1.9937±1.86e+00	**1.5604±1.43e+00(-)**
	10,5	**0.9741±5.90e-01**	1.0048±6.60e-01(+)	**1.3291±9.63e-01**	1.3604±1.21e+00(+)
	5,10	**1.0266±6.49e-01**	1.0595±6.85e-01(+)	**1.5404±1.16e+00**	1.9107±1.62e+00(+)
DF5	10,10	**1.2744±2.08e+00**	1.2906±2.06e+00(+)	**4.4552±6.75e+00**	4.4667±6.51e+00(+)
	10,5	1.3107±2.15e+00	**1.2821±2.90e+00(-)**	4.6213±6.83e+00	**4.3954±6.50e+00(-)**
	5,10	**1.2956±2.03e+00**	1.4014±2.26e+00(+)	**4.6203±6.60e+00**	4.6645±6.83e+00(+)
DF6	10,10	**2.3480±2.87e+00**	2.7282±5.08e+00(+)	**22.2866±2.22e+01**	22.6215±2.51e+01(+)
	10,5	**1.9362±3.61e+00**	2.2292±2.81e+00(+)	**18.4556±1.84e+01**	19.4670±1.85e+01(+)
	5,10	**3.0427±3.25e+00**	3.4270±4.46e+00(+)	**29.2092±3.17e+01**	32.2979±3.96e+01(+)
DF7	10,10	**2.2345±2.32e+00**	2.3662±2.29e+00(+)	**21.5290±2.37e+01**	21.5841±2.39e+01(+)
	10,5	**2.0195±1.48e+00**	2.2923±3.22e+00(+)	**18.9318±2.23e+01**	19.2032±2.05e+01(+)
	5,10	3.1397±4.24e+00	**2.6763±2.63e+00(-)**	**25.8925±3.03e+01**	29.4457±3.22e+01(+)
DF8	10,10	**0.8274±4.58e-01**	1.0325±5.57e-01(+)	0.9117±4.61e-01	**0.8720±4.58e-01(-)**
	10,5	**0.8749±4.29e-01**	1.0616±5.13e-01(+)	0.9506±4.55e-01	**0.9177±4.28e-01(-)**
	5,10	**0.8212±4.69e-01**	1.0332±5.62e-01(+)	0.9684±4.49e-01	**0.9142±4.41e-01(-)**
DF9	10,10	**1.6579±1.69e+00**	1.9618±1.75e+00(+)	4.4558±5.82e+00	**4.3224±5.32e+00(-)**
	10,5	**1.7776±1.77e+00**	2.0620±1.82e+00(+)	**4.9250±6.17e+00**	4.9431±6.57e+00(+)
	5,10	**1.8248±1.83e+00**	1.9418±1.72e+00(+)	**4.6792±5.69e+00**	4.9436±6.67e+00(+)
DF10	10,10	**0.1880±1.34e-01**	0.2139±1.33e-01(+)	0.3571±2.47e-01	**0.3223±2.84e-01(-)**
	10,5	**0.1950±1.39e-01**	0.2634±2.28e-01(+)	**0.3313±2.49e-01**	0.3390±2.41e-01(+)
	5,10	**0.2000±1.12e-01**	0.2465±1.87e-01(+)	0.4822±3.19e-01	**0.4533±3.12e-01(-)**
DF11	10,10	**0.1387±2.31e-02**	0.1455±2.51e-02(+)	**0.4022±1.70e-01**	0.4373±2.24e-01(+)
	10,5	**0.1395±2.27e-02**	0.1456±2.64e-02(+)	**0.3866±1.60e-01**	0.4628±2.45e-01(+)
	5,10	0.1725±3.92e-02	**0.1678±3.88e-02(-)**	**0.4890±2.04e-01**	0.4931±3.38e-01(+)
DF12	10,10	**1.0082±2.45e-01**	1.0767±2.32e-01(+)	**1.0467±2.13e-01**	1.0939±1.97e-01(+)
	10,5	**1.0520±2.14e-01**	1.0898±2.37e-01(+)	**0.9939±2.76e-01**	1.0949±2.32e-01(+)
	5,10	**0.9962±2.68e-01**	1.0525±2.21e-01(+)	**0.9990±2.67e-01**	1.0244±2.59e-01(+)
DF13	10,10	**1.3474±1.87e+00**	1.3716±1.85e+00(+)	4.4537±6.72e+00	**4.3751±6.78e+00(-)**
	10,5	**1.3327±1.81e+00**	1.4081±1.83e+00(+)	4.4694±6.81e+00	**4.3698±6.44e+00(-)**
	5,10	**1.4130±1.93e+00**	1.4151±1.84e+00(+)	**4.5306±6.89e+00**	4.5457±6.76e+00(+)
DF14	10,10	**0.8381±1.38e+00**	0.8673±1.32e+00(+)	**2.9881±4.83e+00**	3.0099±4.99e+00(+)
	10,5	**0.8482±1.37e+00**	0.8660±1.30e+00(+)	3.1810±4.90e+00	**3.0939±4.74e+00(-)**
	5,10	**0.8818±1.38e+00**	0.8903±1.37e+00(+)	3.3921±5.32e+00	**3.1655±4.91e+00(-)**

experimental studies, we investigate the influence of the choice of p on the convergence of KT-MOEA/D.

Fig. 4 shows the mean MIGD values obtained by the KT-MOEA/D algorithms on these test functions. Obviously, KT-MOEA/D with more predicted knee points performs better on most of the test functions, which illustrates more knee points can better guide the population to converge toward the true POF. However, for some test functions which have complicated dynamic POF or POS, such as DF4, DF6, DF7, DF8, and DF10 problems, KT-MOEA/D is difficult to predict the knee points on these test functions. Therefore, the change of p value has little effect on the MIGD values. In addition, on most of the test functions, setting more than ten knee points brings little improvement on convergence. It is the reason for setting p to 10 in this article.

E. Sensitivity Study

The main purpose of KT-MOEA/D is to reuse acquired knowledge by applying prediction and imbalanced transfer learning techniques. To validate the proposed design components, we quantify the performance of KT-MOEA/D and KT-MOEA/D$_\sigma$, which predict without imbalanced transfer learning techniques. In the KT-MOEA/D$_\sigma$, the factor σ is fixed as 1, it means the weights in the source domain will keep a fast convergence, and these minority solutions have little impact on the model training. Table III and Table S-IV in the supplementary record the MIGD metric values and the MHV metric values of KT-MOEA/D and KT-MOEA/D$_\sigma$, respectively.

As can be seen from Table III, KT-MOEA/D achieves 34 out of 42 best MIGD metric values when the number of decision variables is 10. On the other hand, when the number of decision variables is 30, KT-MOEA/D achieves 25 best result. For MHV metric values, Table S-IV in the supplementary material shows KT-MOEA/D performs better on 28 out of the 42 problems when the number of decision variables is 10. On the other hand, when the number of decision variables is 30, KT-MOEA/D achieves 27 best results.

The experimental results illustrate that the KT-MOEA/D with dynamic factor σ can achieve better performance than KT-MOEA/D$_\sigma$ does. The reason is that the number of knee points is very low in the training set. When the data are imbalanced, the small number of samples is likely to be misclassified by the weak classifiers, which leads to faster convergence of the weights in the source domain. Thus, introducing the factor σ can prevent fast decreasing of the weight values of samples from the minority class, and the performance of prediction is improved.

V. Conclusion

Using transfer learning techniques to solve DMOPs has proven to be a promising approach [9]. However, the existing methods tend to be slower, which limits the scope of application of such methods. One of the reasons for the slow running speed is that low-quality individuals consume a large amount of computing resources, and the under-performance of these individuals may lead to negative transfer. Solely exploiting high quality individuals, such as knee points, with transfer learning is considered a better solution to this problem. In this article, a knee-point-based imbalanced transfer learning method, called KT-DMOEA, was proposed for solving DMOPs. The proposed method divides the whole decision space into several regions, and a prediction model, TPM, generates an estimated knee point for each region on interest. However, the classic transfer learning algorithm cannot be used directly to obtain the initial population for the optimization function at the next time step since the number of knee points is very small. To solve this problem, we propose an imbalanced transfer learning method, KT-DMOEA, that attend to the training class imbalance problem by adjusting the weights of the solutions. The extensive experiments validate that the proposed prediction approach is more accurate than some chosen state-of-the-art algorithms. In our future work, we will continue to explore how to reduce the possibility of negative transfer and propose more efficient DMOPs algorithms.

References

[1] M. Jiang, Z. Huang, G. Jiang, M. Shi, and X. Zeng, "Motion generation of multi-legged robot in complex terrains by using estimation of distribution algorithm," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, 2017, pp. 1–6.

[2] W. Du, W. Zhong, Y. Tang, W. Du, and Y. Jin, "High-dimensional robust multi-objective optimization for order scheduling: A decision variable classification approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 293–304, Jan 2019.

[3] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 6, pp. 1–24, Oct. 2012.

[4] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 838–856, Dec. 2015.

[5] Z. Zhan *et al.*, "Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 704–716, Mar. 2017.

[6] C. Li, T. T. Nguyen, M. Yang, M. Mavrovouniotis, and S. Yang, "An adaptive multipopulation framework for locating and tracking multiple optima," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 590–605, Aug. 2016.

[7] Y. Wang, Z.-Z. Liu, J. Li, H.-X. Li, and G. G. Yen, "Utilizing cumulative population distribution information in differential evolution," *Appl. Soft Comput.*, vol. 48, pp. 329–346, Nov. 2016.

[8] Y. Jin, C. Yang, J. Ding, and T. Chai, "Reference point based prediction for evolutionary dynamic multiobjective optimization," in *Proc. Evol. Comput.*, 2016, pp. 3769–3776.

[9] 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.

[10] M. Jiang, L. Qiu, Z. Huang, and G. G. Yen, "Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation," *Inf. Sci.*, vol. 435, pp. 203–223, Apr. 2018.

[11] A. T. W. Min, Y. Ong, A. Gupta, and C. Goh, "Multiproblem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 15–28, Feb. 2019.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[13] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, Dec. 2015.

[14] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[15] Z. He and G. G. Yen, "Many-objective evolutionary algorithms based on coordinated selection strategy," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 220–233, Apr. 2017.

[16] Q. Li, J. Zou, S. Yang, J. Zheng, and R. Gan, "A predictive strategy based on special points for evolutionary dynamic multi-objective optimization," *Soft Comput.*, vol. 23, no. 1, pp. 1–17, 2018.

[17] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Parallel Problem Solving from Nature—PPSN VIII*, X. Yao *et al.*, Eds. Berlin, Germany: Springer, 2004, pp. 722–731.

[18] I. Das, "On characterizing the 'knee' of the Pareto curve based on normal-boundary intersection," *Struct. Optim.*, vol. 18, no. 2, pp. 107–115, Oct. 1999.

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

[20] K. Deb, B. R. N. Udaya, and S. Karthik, *Dynamic Multi-Objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-Thermal Power Scheduling*. Heidelberg, Germany: Springer, 2007.

[21] D. Yazdani, M. N. Omidvar, J. Branke, T. T. Nguyen, and X. Yao, "Scaling up dynamic optimization problems: A divide-and-conquer approach," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 1–15, Feb. 2020.

[22] S. Zeng, R. Jiao, C. Li, X. Li, and J. S. Alkasassbeh, "A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2678–2688, Sep. 2017.

[23] 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.

[24] 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.

[25] R. Azzouz, S. Bechikh, and L. B. Said, "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy," *Soft Comput.*, vol. 21, no. 4, pp. 885–906, 2017.

[26] M. Jiang, W. Hu, L. Qiu, M. Shi, and K. C. Tan, "Solving dynamic multi-objective optimization problems via support vector machine," in *Proc. 10th Int. Conf. Adv. Comput. Intell. (ICACI)*, Mar. 2018, pp. 819–824.

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

[28] 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.

[29] 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.

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

[31] Z. Peng, J. Zheng, J. Zou, and M. Liu, "Novel prediction and memory strategies for dynamic multiobjective optimization," *Soft Comput.*, vol. 19, no. 9, pp. 2633–2653, Sep. 2015.

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

[33] 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.

[34] C. Bu, W. Luo, and L. Yue, “Continuous dynamic constrained optimization with ensemble of locating and tracking feasible regions strategies,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 14–33, Feb. 2017.

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

[36] M. Rong, D. Gong, W. Pedrycz, and L. Wang, “A multi-model prediction method for dynamic multi-objective evolutionary optimization,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 290–304, Apr. 2020.

[37] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, “Boosting for transfer learning,” in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 193–200.

[38] B. Krawczyk, “Learning from imbalanced data: Open challenges and future directions,” *Progr. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.

[39] 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.

[40] 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.

[41] S. Jiang, S. Yang, X. Yao, K. Tan, M. Kaiser, and N. Krasnogor, “Benchmark problems for CEC2018 competition on dynamic multiobjective optimisation,” in *Proc. CEC Competition*, 2018, pp. 1–18.

[42] 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.

[43] Q. Zhang, A. Zhou, and Y. Jin, “RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.

[44] L. While, P. Hingston, L. Barone, and S. Huband, “A faster algorithm for calculating hypervolume,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.

[45] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, May 2011.

[46] 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.

Min Jiang (Senior Member, IEEE) received the bachelor’s and Ph.D. degrees in computer science from Wuhan University, Wuhan, China, in 2001 and 2007, respectively.

Subsequently as a Postdoctoral Researcher with the Department of Mathematics, Xiamen University, Xiamen, China, where he is currently a Professor with the Department of Artificial Intelligence. His main research interests are machine learning, computational intelligence, and robotics. He has a special interest in dynamic multiobjective optimization, transfer learning, the software development, and in the basic theories of robotics.

Dr. Jiang received the Outstanding Reviewer Award from the IEEE TRANSACTIONS ON CYBERNETICS in 2016. He is currently serving as Associate Editors for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and he is the Chair of IEEE CIS Xiamen Chapter.

Zhenzhong Wang received the bachelor’s degree in computer science and technology from Northeastern University, Shenyang, China, in 2017. He is currently pursuing the master’s degree with the School of Informatics, Xiamen University, Xiamen, China.

His research interests include computational intelligence and machine learning.

Haokai Hong received the bachelor’s degree in computer science from Xiamen University, Xiamen, China, in 2020, where he is currently pursuing the master’s degree with the School of Informatics.

His research interests include computational intelligence and machine learning.

Gary G. Yen (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Notre Dame, Notre Dame, IN, USA, in 1992.

In 1997, he was with the Structure Control Division, U.S. Air Force Research Laboratory, Albuquerque, NM, USA. He is currently a Regents Professor with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, USA. His research interests include intelligent control, computational intelligence, conditional health monitoring, signal processing, and their industrial/defense applications.

Dr. Yen received the Andrew P Sage Best Transactions Paper Award from the IEEE Systems, Man, and Cybernetics Society in 2011, and the Meritorious Service Award from IEEE Computational Intelligence Society in 2014. He was an Associate Editor of the *IEEE Control Systems Magazine*, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, *Automatica*, *Mechantronics*, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, and the IEEE TRANSACTIONS ON NEURAL NETWORKS. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS. He served as the General Chair for the 2003 IEEE International Symposium on Intelligent Control held in Houston, TX, USA, and 2006 IEEE World Congress on Computational Intelligence held in Vancouver, BC, Canada. He served as a Vice President for the Technical Activities from 2005 to 2006, a President of the IEEE Computational Intelligence Society from 2010 to 2011, and is the Founding Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2006 to 2009. He is a fellow of IET.