Tracking Moving Optima of Dynamic Multi-objective Problem via Prediction in Objective Space

Wei Zhou[1], Liang Feng[1], Zexuan Zhu[2], Kai Liu[1], Chao Chen[1], Zhou Wu[3]

College of Computer Science, Chongqing University, China[1]
College of Computer Science and Software Engineering, Shenzhen University, China[2]
College of Automation, Chongqing University, China[3]
Email: {jerryzhou, liangf}@cqu.edu.cn

***Abstract*—Solving dynamic multi-objective optimization problem (DMOP) requires optimizing multiple conflicting objectives simultaneously. When a dynamic is detected in the changing environment, most of existing prediction-based strategies predict the trajectory of changing Pareto-optimal solutions (POS), based on the historical solutions obtained in the solution space. In this paper, we present a new prediction method to track the moving optima for solving DMOP. In contrast to existing approaches, we propose to build the prediction model in the objective space. As the evaluation for solving a DMOP is based on the Pareto-optimal front (POF), to predict directly in the objective space could provide more useful information than the prediction in the solution space. In particular, to efficiently capture the complex relationships among POFs found along the evolutionary search, here we build a prediction model in Reproducing Kernel Hilbert Space, which holds a closed-form solution. To evaluate the performance of the proposed method, empirical studies have been conducted by comparing against three state-of-the-art prediction-based strategies on fourteen commonly used DMOP benchmarks. The results obtained by using different optimization solvers confirmed the superiority of the proposed method for solving DMOP in terms of both solution quality and time efficiency.**

***Index Terms*—Evolutionary Optimization, Dynamic Multi-objective Optimization Problem, Prediction in Objective Space**

I. Introduction

Multi-objective optimization problem (MOP) generally involves at least two conflicting objectives to be optimized simultaneously [1]. Unlike single objective optimization, no single solution can satisfy a given MOP, thus the optima of MOP are a set of solutions in the decision space, namely Pareto-optimal solutions (POS) [2]. The objective values of the POS in the objective space are known as the Pareto-optimal front (POF).

In the literature, a large number of multi-objective evolutionary algorithms (MOEAs) have been proposed to solve MOP, which are capable of approximating the POS or POF efficiently in the static environment [3], [4], [5]. However, the MOPs in real-world applications are usually dynamic, which possess POS and POF changing over time. Particular instances of dynamic multi-objective optimization problems (DMOPs) can be found in applications in control [6], dynamic scheduling [7], resource management [8], and routing [9], [10], etc. Static MOPs are commonly optimized in a given time budget with static problem properties, while the optimization of DMOPs is more challenging due to the time-varying POS or POF [11], [12]. Therefore, MOEAs for tackling DMOPs (DMOEAs) require the design of additional strategies to react to the changes, such as introducing or maintaining diversity [13], [14], change prediction [15], [16], [17], and extra memory design [18], [19]. As most DMOPs exhibit certain predictable patterns in the moving of POS or POF, among existing DMOEAs, prediction-based approaches generally achieve superior optimization performance for solving DMOPs in contrast to the others.

To track the moving directions of optima in a given DMOP, the correlations between sequentially non-dominated solutions obtained along the evolutionary search are usually used to build the prediction model of the DMOP [20]. In particular, most of existing prediction methods further simplify the relationship between solutions in two sequential time periods as linear correlation. Therefore, linear prediction models are trained to predict the moving POS of a given DMOP, such as autoregressive model [15], [21], linear predictor [22], [23], [24] and Kalman filter [16]. Furthermore, as nonlinearity may exist in the POS solutions found along the search, nonlinear predictor has also been proposed in the literature. For instance, Cao *et al.* [20] proposed a Support Vector Regression (SVR) predictor combined with MOEA/D, to capture diverse types of correlations existed between the obtained POS solutions over time. In both the linear and nonlinear approaches, it is worth noting that they all build the prediction models in the solution space using the optimized POS solutions. However, as the evaluation in solving a given DMOP is based on the objective values, the POF found while the search progresses online could provide more accurate information in contrast to the POS. In the literature, to the best of our knowledge, there is only little work conducted to explore the prediction model for solving DMOP in the objective space. In particular, Jiang *et al.* in [17] presented a transfer learning approach based on transfer component analysis (TCA) to predict the moving of optima, using POFs obtained in two consecutive time instances. Nevertheless, as it requires to build common representation of different POFs periodically, it is computational expensive and thus could fail to provide high quality prediction for solving DMOP when limited computational budget is given.

Keeping the above in mind, in this paper, we propose a new prediction method to tackle DMOPs using information obtained in the objective space while the search progresses online. In particular, with respect to both accuracy and efficiency of the prediction, we propose to build the prediction model in Reproducing Kernel Hilbert Space to track the moving of POF in solving DMOP. The proposed method holds a closed-form solution, which thus makes the prediction extremely efficient in reacting to the dynamic changes. Moreover, the proposed method is designed to regenerate an initial population for the search in a new environment, which can be easily integrated into any population-based optimization algorithms for solving DMOPs. To evaluate the efficacy of the proposed method, empirical studies are conducted on commonly used DMOP benchmarks against three recently proposed DMOEAs based on different evolutionary optimizers. The obtained results confirmed the efficiency and effectiveness of the proposed prediction method for solving DMOPs.

The rest of this paper is organized as follows. The definition of the DMOP studied in this paper as well as some related work of prediction-based approaches are presented in Section II. In Section III, we detail the design of the proposed prediction method for evolutionary dynamic multi-objective optimization. Subsequently, Section IV provides the empirical studies on the IEEE CEC2018 DMOP benchmarks, by comparing the proposed method with three state-of-the-art prediction-based DMOEAs. Lastly, we conclude this work and discuss our future work in Section V.

II. Preliminaries

In this section, the background of the DMOP is introduced firstly. Next, a brief review of the prediction-based approaches is also presented.

A. Dynamic Multi-objective Optimization Problem

In this paper, we consider the DMOP as a time-variant MOP. In particular, minimization problems are investigated here. Formally, the mathematical definition of a DMOP is formulated as [11], [20]:

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

where $\Omega \subset \mathbb{R}^n$ represents the decision (variable) space, and $\mathrm{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is a vector of n-number of decision variables in this space. The given DMOP involves an m-dimensional objective space $\mathbb{R}^m$ that comprises m-number of objective functions, while f_i denotes the i-th objective function, which is continuous with respect to x, x_i is in a range between L_i and U_i. $F(\mathrm{x}, t)$ is the set of objective functions with respect to time (or other dynamics) of the problem, which is represented by t.

At the time of index t, a decision vector x_1 is Pareto dominated by another vector x_2, denoted by $\mathrm{x}_1 \prec_t \mathrm{x}_2$, if and only if:

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

Based on the concept of dynamic Pareto dominance, the definitions of DPOS and DPOF are given below:

Definition 1 (Dynamic Pareto Optimal Solutions)**.**

The POS at time t, denoted as $DPOS(t)^*$, is the set of all Pareto optimal solutions in the decision space such that:

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

Definition 2 (Dynamic Pareto Optimal Front)**.**

The POF at time t, denoted as $DPOF(t)^*$, is the corresponding projection of the $DPOS(t)^*$ in the objective space such that:

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

B. Prediction-based Approaches in Dynamic Multi-objective Optimization

In recent years, many research attempts have been made in the design of prediction-based DMOEAs for solving DMOPs. As the behavior of the dynamic often follows a certain trend, prediction-based approaches can exploit the relationship between solutions obtained from two consecutive time instances, thus to track the changing POS or POF over time. Without loss of generality, a framework following the prediction-based approaches is presented in Algorithm. 1. More precisely, the common procedure of population prediction is as follows. Firstly, to estimate the states of the changing environment

Algorithm 1 General Prediction-based Approach

```
1. Initialization:
      a. Initialize the population;
      b. Initialize the learning model and training set.
2. Search for optima
3. Change detection:
      a. Re-evaluate dedicated detectors;
      b. Assess algorithm behaviors.
4. Population prediction (If a change is detected in Step 3):
      a. Configure the obtained environment state as the input of the learning model;
      b. Use the learning model to estimate the type of this current change and/or the next change;
      c. Generate new individuals as initial population that best match with the estimation.
5. Return to Step 2 and update the training set.
```

by utilizing the obtained optimization knowledge with some forms of machine learning techniques, then to predict new individuals as change reaction such that the DMOEA can adapt itself to changes for tackling DMOPs. In contrast to the flow of static MOEAs, *Change detection* and *Population prediction* are two additional elements in prediction-based DMOEAs. In the literature, most of prediction methods are proposed with a combination of *Change detection* and *Population prediction* for solving DMOPs, wherein the research generally focused on the design of prediction. In this section, some representative as well as recent prediction methods are reviewed below.

Feedforward prediction strategy (FPS) proposed in [21] utilizes an autoregressive (AR) forecasting model to predict some isolated individuals of the new population. Further in [15], Population Prediction Strategy (PPS) presented by Zhou *et al.* also adopted the AR model, to predict the whole population of MOEA. The proposed PPS mainly contains two parts: center point prediction and manifold prediction. This strategy maintained a sequence of preceding center points to predict the new center by the AR model, and the new manifold is estimated by the preceding manifolds, thus to form a whole population. More recently, Rong *et al.* [25] proposed a multidirectional prediction strategy (MDP) to enhance the performance of MOEAs in solving DMOP. Instead of using the center point, a number of individuals, which can describe the location and the diversity of the POS, were used to track the moving of POS. In summary, the above approaches conduct the *Population prediction* focusing on the change of some deterministic data points, such as center point, knee point and boundary point, etc., which is not able to adequately capture the changes of the whole POS.

In contrast to the strategies using deterministic knowledge in *Population prediction*, Muruganantham *et al.* proposed a new prediction model in [16] that combines linear Kalman filter (KF) prediction and a scoring scheme. The KF assumes that the decision variables are independent and helps to guide the search towards the changing optima; thereby, the DMOEA can quickly track the moving optima in the solution space. To overcome the limitation caused by the IID hypothesis in DMOPs, Jiang *et al.* [17] introduced the integration of TCA and MOEAs to explore the prediction for solving DMOP in objective space, which has properly addressed the violation of KF assumption that the relationship between solutions is linear. However, heavy computational burden is brought to the transfer learning based prediction, due to iterative domain adaptation learning applied in the search process.

III. Proposed Method

In this section, the details of the proposed prediction method for evolutionary dynamic multi-objective optimization are presented. The whole flowchart of the proposed algorithm follows Algorithm. 1. In particular, by employing a conventional MOEA as the basic optimization solver, the search process starts as routine, following the procedures as in static environment. However, an additional detection operator is designed to trigger the prediction operator when a change is detected. In what follows, we first introduce the environmental-change detection operator employed in the proposed method, then the details of the prediction operator, and the overall framework of the DMOEA based on prediction in objective space (OP-DMOEA) are elaborated.

A. Dynamic Detection

Before entering into the phase of population prediction, a signal of change shall be known and give notice to the algorithm [26]. In this paper, DMOPs are tackled in the unknown environment, it is crucial for the DMOEA to detect changes along the search process. In the literature, according to the framework of prediction-based methods outlined in Section II-B, existing dynamic detection methods can be categorized into two groups, i.e., detector-based detection and behavior-based detection. In particular, the detector-based methods re-evaluate some specific solutions (detectors) to detect changes in their objective values or feasibilities, while the behavior-based approaches consider to assess the behaviors of algorithm for dynamic detection.

Following the robust detection performance achieved in previous DMOEAs [16], [25], [27], herein, we do not propose a new solution for the design of detection operator, while a simple yet effective detector-based detection method is employed. Particularly, to reduce the number of function evaluations and maintain the coverage of the detection region, we randomly select 5% individuals as detectors in the population, and store their objective values. At the start of each generation, the detectors will be re-evaluated, when the current objective values of these detectors vary from the stored ones at last generation, it is recognized that an change occurs. If no change is detected, the algorithm performs static optimization and evolves the population.

B. Proposed prediction method

In the context of solving DMOP, prediction methods aim at learning the predictable patterns exhibited in solutions collected from the preceding environments. However, with the learning of linear mapping between the optimization data obtained in two consecutive time instances, nonlinearity is hardly captured. In order to address this issue, we propose a new solution to construct a prediction model with closed-form solution in Reproducing Kernel Hilbert Space (RKHS).

In particular, we denote the optimization problems of a DMOP before and after the dynamic occurs as $\mathbf{OP}_1$ and $\mathbf{OP}_2$, respectively. The solutions of these two optimization problems are represented by $\mathbf{P} \in \mathbb{R}^{d\times N}$ and $\mathbf{Q} \in \mathbb{R}^{d\times N}$, respectively, i.e., $\mathbf{P} = \{p_1 \ldots, p_N\}$ and $\mathbf{Q} = \{q_1 \ldots, q_N\}$, where N denotes the number of solutions in each set, and d is the dimension. Naturally, through learning the mapping $\mathbf{M} \in \mathbb{R}^{d\times d}$ from $\mathbf{P}$ to $\mathbf{Q}$ [28], the connection between $\mathbf{OP}_1$ and $\mathbf{OP}_2$ is also built by $\mathbf{M}$.

Apply kernelization in the learning of $\mathbf{M}$, suppose that $\mathbf{P}$ is mapped to RKHS $\mathcal{H}$ by a nonlinear mapping function $\Phi : \mathbb{R}^d \to \mathcal{H}$. Denote $\Phi(\mathbf{P}) = [\Phi(p_1, \ldots, \Phi(p_N)]$ as the mapped data matrix. The original loss function in [28], which serves to build connection across problems, thus can be rewritten with kernelized terms as:

$$L(\mathbf{M}) = \frac{1}{2N} tr[(\mathbf{Q} - \mathbf{M}\Phi(\mathbf{P}))^{\mathrm{T}}(\mathbf{Q} - \mathbf{M}\Phi(\mathbf{P}))] \tag{5}$$

where T is the transpose operation of a matrix. According to [29], the linear mapping $\mathbf{M}$ can be represented as a linear combination of the mapped data points in $\mathcal{H}$, that is, $\mathbf{M} = \mathbf{M}_k\Phi(\mathbf{P})^{\mathrm{T}}$, and the loss minimization function becomes:

$$L(\mathbf{M}_k) = \frac{1}{2N} tr[(\mathbf{Q} - \mathbf{M}_k\mathbf{K}(\mathbf{P},\mathbf{P}))^{\mathrm{T}}(\mathbf{Q} - \mathbf{M}_k\mathbf{K}(\mathbf{P},\mathbf{P}))] \tag{6}$$

where $\mathbf{K}(\mathbf{P},\mathbf{P}) = \Phi(\mathbf{P})^{\mathrm{T}}\Phi(\mathbf{P})$ is the kernel matrix with $k_{i,j} = k(p_i, p_j)$, and $k(\cdot,\cdot)$ is a kernel function. Eq. 6 holds a closed-form solution, which is given by:

$$\begin{aligned} &\mathbf{M}_k = \mathbf{P}_k(\mathbf{Q}_k)^{-1}\text{with} \\ &\mathbf{P}_k = \mathbf{Q}\mathbf{K}(\mathbf{P},\mathbf{P})^{\mathrm{T}}, \mathbf{Q}_k = \mathbf{K}(\mathbf{P},\mathbf{P})\mathbf{K}(\mathbf{P},\mathbf{P})^{\mathrm{T}} \end{aligned} \tag{7}$$

As can be observed in Eq. 6, the connection between two optimization problems is built by the learned matrix $\mathbf{M}_k$. Moreover, the relationships among the optimization data can be simply captured by the multiplication of $\mathbf{M}_k$ and the corresponding kernel matrix.

C. Framework of The Proposed OP-DMOEA

When environmental change is detected in solving a given DMOP, the proposed prediction operator receives the non-dominated solutions from the evolutionary search process of a chosen MOEA, and the prediction of subsequent DPOF is triggered at the beginning of the third time period. We assume a sequence of non-dominated solutions found by the basic optimization solver in the previous time periods are denoted as $\mathrm{NDS} = \{\mathrm{NDS}_0, \mathrm{NDS}_1, \ldots, \mathrm{NDS}_{t-1}, \mathrm{NDS}_t\}$. Herein, we propose to utilize the DPOFs obtained in the time windows $t-1$ and t, to learn the moving directions of DPOF. With the learned moving directions, a set of objective values is estimated. The overall framework of the proposed OP-DMOEA is presented in Algorithm. 2.

In particular, We first sort the solutions of NDS_{t-1} and NDS_t independently, with respect to the ranking indicator corresponding to the basic MOEA. Next, calculate their objective values via objective functions $F(\mathrm{NDS}_{t-1}, t-1)$ and $F(\mathrm{NDS}_t, t)$ and archive the values into $DPOF(t-1)$ and $DPOF(t)$, respectively. Further, by configuring $DPOF(t-1)$ and $DPOF(t)$, as the input $\mathbf{P}$ and output $\mathbf{Q}$ of the learning model discussed in Section III-B, respectively, the moving directions of DPOF from time window $t-1$ to t can be modeled by the matrix $\mathbf{M}_k$ learned via Eq. 7. To predict the objective values in time window $t+1$ (i.e., Y_{t+1}), new kernel matrix with a determined kernel function is derived to multiply with the corresponding mapping matrix $\mathbf{M}_k$, which is formulated as follows:

$$\begin{aligned} \mathrm{Y}_{t+1} &= \mathbf{M}\Phi(DPOF(t)) \\ &= \mathbf{M}_k\Phi(DPOF(t-1))^{\mathrm{T}}\Phi(DPOF(t)) \\ &= \mathbf{M}_k\mathbf{K}(DPOF(t-1), DPOF(t) \end{aligned} \tag{8}$$

Algorithm 2 OP-DMOEA: Dynamic Multi-objective Evolutionary Algorithm via Prediction in Objective Space

```
Input: $F(\mathrm{x},\cdot)$: the dynamic objective function; a multi-objective evolutionary algorithm MOEA.
Output: $DPOF$: the DPOFs of the DMOP.
1:  Initialization;
2:  Solve $F(\mathrm{x},0)$ and $F(\mathrm{x},1)$ with the MOEA to get $\mathrm{NDS}_0$ and $\mathrm{NDS}_1$, respectively;
3:  Rank solutions in $\mathrm{NDS}_0$ and $\mathrm{NDS}_1$;
4:  Calculate the $DPOF(0) = F(\mathrm{NDS}_0, 0)$ and $DPOF(1) = F(\mathrm{NDS}_1, 1)$, respectively, and randomly select 5% individuals in $\mathrm{NDS}_1$ as detectors and archive their objective values;
5:  Set time index $t = 1$;
6:  $DPOF = DPOF(0) \cup DPOF(1)$
7:  for $t = 1$ to $end$ do
8:      Configure the input $\mathbf{P}$ and output $\mathbf{Q}$ of the learning model, as $DPOF(t-1)$ and $DPOF(t)$, respectively;
9:      Obtain the matrix $\mathbf{M}_k$ via Eq. 7;
10:     Predict initial population $\mathrm{IP}_{t+1}$ via Eq. 8 and Eq. 9;
11:     Search for $\mathrm{NDS}_{t+1}$ and calculate the $DPOF(t+1)$;
12:     $DPOF = DPOF \cup DPOF(t+1)$
13: end for
14: return the $DPOF$.
```

Since the predicted Y_{t+1} is a set of objective values, an additional mapping function is required to map these values back into solution space as initial population for the search in time window $t+1$ (i.e., IP_{t+1}). The objective value of newly found individual in the initial population is the closest with respect to the predicted value. Hence, to locate the individuals in the solution space, a single objective optimization problem need to be solved here. With a predicted value y^* in Y_{t+1}, an initial solution x^* is calculated as:

$$x^* \leftarrow \min \|\mathrm{F}(x^*, t+1) - y^*\| \tag{9}$$

We follows the adoption of the Interior Point Algorithm in [17] to solve this problem.

Last but not the least, the predicted solutions compose a new population as change reaction, thus to accelerate the search toward the true POF in the new environment. Observing this, the proposed prediction method is optimizer independent, it thus can be easily integrated into any of population-based static MOEAs for tackling DMOPs. In the proposed prediction operator, the Gaussian RBF kernel is employed as the kernel function.

IV. Empirical Study

In this section, to evaluate the performance of the proposed OP-DMOEA for solving DMOPs, empirical studies on commonly used DMOP benchmarks against three recently proposed DMOEAs are presented.

A. Experimental Setup

1) Test Instances: The IEEE CEC2018 Competition on Dynamic Multi-objective Optimization Benchmark problems [30]

TABLE I
MEAN AND STANDARD DEVIATIONS OF MIGD VALUES OBTAINED BY THE COMPARED ALGORITHMS ON DMOPS POSSESSING TWO OBJECTIVES

		MOEA/D Optimizer		*MOPSO Optimizer*		
Problems	(n_t,τ_t)	*Tr-MOEA/D*	*OP-MOEA/D*	*MDP-MOPSO*	*Tr-MOPSO*	*OP-MOPSO*
DF1	(5,10)	**1.9955E-02 (1.9053E-02)**≈	2.7862E-02 (2.7104E-02)	6.2201E-02 (9.8853E-03)+	2.7879E-02 (1.0109E-02)≈	**2.4812E-02 (1.0578E-02)**
	(2.5,10)	4.5946E-02 (2.5566E-02)≈	**4.4469E-02 (1.9279E-02)**	2.3505E-01 (2.5293E-02)+	**3.8378E-02 (1.0324E-02)–**	7.3248E-02 (3.5875E-02)
	(1,10)	5.8406E-02 (1.3437E-02)≈	**5.0459E-02 (1.0633E-02)**	1.8502E+00 (2.9654E-01)+	**3.3164E-02 (1.6444E-02)–**	1.5958E-01 (7.0041E-02)
DF2	(5,10)	**2.3551E-02 (1.0912E-02)**≈	2.5035E-02 (2.2909E-02)	9.7943E-02 (2.0205E-02)+	**1.9316E-02 (6.9590E-03)–**	2.9791E-02 (7.1038E-03)
	(2.5,10)	6.4191E-02 (7.4648E-02)+	**3.2250E-02 (2.1324E-02)**	1.8657E-01 (1.8058E-02)+	**1.8152E-02 (6.8592E-03)–**	3.7037E-02 (1.1480E-02)
	(1,10)	5.9982E-02 (2.9287E-02)≈	**5.8742E-02 (2.5481E-02)**	1.5970E+00 (1.5952E-01)+	**1.6540E-02 (7.9081E-03)–**	1.5934E-01 (6.5745E-02)
DF3	(5,10)	2.6010E-01 (1.8158E-02)≈	**2.3318E-01 (8.1696E-02)**	3.6925E-01 (3.0446E-02)+	1.2284E+00 (4.7810E-01)+	**2.8142E-01 (4.6781E-02)**
	(2.5,10)	2.1170E-01 (1.8949E-02)+	**1.7948E-01 (5.8183E-02)**	4.4430E-01 (3.0045E-02)+	1.0622E+00 (3.3991E-01)+	**2.9103E-01 (4.2715E-02)**
	(1,10)	1.5329E-01 (2.2963E-02)≈	**1.4346E-01 (3.3503E-02)**	4.7208E-01 (6.5452E-02)+	8.0173E-01 (3.6145E-01)+	**3.1774E-01 (7.7236E-02)**
DF4	(5,10)	**1.2311E-01 (2.4970E-02)**≈	1.2984E-01 (3.0900E-02)	1.0516E+00 (1.4857E-01)+	1.1962E+00 (2.1267E-01)+	**3.8251E-01 (9.4019E-02)**
	(2.5,10)	1.1015E-01 (1.9350E-02)≈	**1.0516E-01 (1.6974E-02)**	1.0946E+00 (1.0628E-01)+	1.1415E+00 (2.4204E-01)+	**3.4825E-01 (8.9705E-02)**
	(1,10)	6.1115E-02 (9.4059E-03)≈	**5.9841E-02 (2.2882E-02)**	1.2851E+00 (1.3134E-01)+	1.3199E+00 (1.7648E-01)+	**2.6830E-01 (7.0662E-02)**
DF5	(5,10)	1.6426E-02 (8.6040E-03)+	**1.1099E-02 (2.2385E-03)**	1.3096E-01 (1.8647E-02)+	3.2623E-01 (7.3027E-02)+	**7.2574E-02 (6.6264E-02)**
	(2.5,10)	1.3889E-02 (1.5163E-03)+	**1.1440E-02 (2.9452E-03)**	1.8084E-01 (1.2646E-02)+	5.1224E-01 (1.4839E-01)+	**8.7710E-02 (7.3333E-02)**
	(1,10)	3.8425E-02 (1.0211E-02)+	**1.3905E-02 (3.2474E-03)**	3.2212E-01 (1.0808E-01)+	8.5409E-01 (1.0320E-01)+	**5.9112E-02 (4.6411E-02)**
DF6	(5,10)	1.2814E+00 (3.9050E-01)+	**8.6064E-01 (3.5330E-01)**	**2.3972E+00 (3.4955E-01)–**	1.2051E+01 (1.1314E+00)+	3.7035E+00 (6.0488E-01)
	(2.5,10)	5.7628E-01 (5.3989E-02)+	**3.0538E-01 (5.0614E-02)**	**1.9676E+00 (3.9709E-01)**≈	9.7248E+00 (7.0239E-01)+	2.0857E+00 (3.8609E-01)
	(1,10)	4.4211E-01 (2.2433E-02)+	**1.8347E-01 (1.9260E-02)**	1.0413E+00 (4.7044E-01)≈	7.6502E+00 (9.4631E-01)+	**8.9059E-01 (3.1032E-01)**
DF7	(5,10)	**2.6546E-01 (4.1794E-02)**≈	4.3349E-01 (1.5637E-01)	3.2899E-01 (3.6425E-02)–	**2.7422E-01 (3.4569E-02)–**	7.7691E-01 (3.6755E-02)
	(2.5,10)	**7.2006E-01 (9.5435E-02)–**	1.2066E+00 (4.6182E-02)	5.7898E-01 (6.3800E-02)–	**4.5970E-01 (7.1846E-02)–**	1.3340E+00 (5.7692E-02)
	(1,10)	**1.8238E+00 (3.2338E-01)–**	2.9086E+00 (1.4497E-01)	**7.6362E-01 (6.2217E-02)–**	8.3281E-01 (8.6814E-02)–	2.3489E+00 (1.7624E-01)
DF8	(5,10)	**1.0486E-01 (1.6498E-02)–**	2.5655E-01 (2.2962E-02)	2.2670E-01 (9.7179E-03)+	3.3067E-01 (5.0169E-02)+	**1.8595E-01 (2.3775E-02)**
	(2.5,10)	**1.2650E-01 (4.2268E-02)–**	2.2443E-01 (2.9958E-02)	2.5663E-01 (9.0549E-03)+	3.5144E-01 (4.9894E-02)+	**1.5102E-01 (1.2471E-02)**
	(1,10)	**1.0764E-01 (3.7231E-02)–**	2.4674E-01 (5.7031E-02)	2.2199E-01 (1.2884E-02)+	1.9679E-01 (1.2299E-02)+	**1.0615E-01 (8.4044E-03)**
DF9	(5,10)	6.5768E-02 (9.3095E-03)+	**3.4521E-02 (9.2937E-03)**	4.0295E-01 (5.1122E-02)+	8.8193E-01 (1.4022E-01)+	**1.8431E-01 (2.1593E-02)**
	(2.5,10)	6.1455E-02 (8.3039E-03)+	**3.3885E-02 (9.2815E-03)**	5.0768E-01 (3.5292E-02)+	9.4988E-01 (2.0971E-01)+	**2.1985E-01 (2.7217E-02)**
	(1,10)	8.2888E-02 (2.9222E-02)+	**3.0782E-02 (4.2557E-03)**	4.8227E-01 (2.8609E-02)+	9.9123E-01 (7.9726E-02)+	**2.1740E-01 (1.8893E-02)**

Superior performances are highlighted in bold. "+", "–", and "≈" denote proposed OP-DMOEA is statistically significant better, worse, and similar to the compared dynamic multi-objective method which shares the same multi-objective optimizer, respectively.

are used to evaluate our proposed OP-DMOEA. The test suite (called DF in this competition) has fourteen test problems, including nine two-objective and five three-objective problems, whose dynamic characteristics cover diverse properties representing various real-world scenarios [20]. The dynamics of these test functions are governed by a time variable (t), which is defined as:

$$t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor, \tag{10}$$

where τ is the generation counter and $\lfloor \cdot \rfloor$ is the floor operator. A smaller value of n_t means larger changes, while a smaller value of τ_t leads to more frequent occurrence of changes in the DMOP. Towards comprehensive experiments, three different configurations of dynamic changes are investigated in the following experiments, which are $(n_t, \tau_t) = (5, 10), (2.5, 10)$ and $(1, 10)$.

2) Performance Metrics: In this paper, we employ two metrics, the Inverted Generational Distance (IGD) [31] and one of its variants, to measure the quality of obtained solutions and to assess the performance of the competing algorithms.

IGD is a performance indicator which quantify the solution quality of a MOEA. It is mathematically given by:

$$\text{IGD}(POF^{t*}, POF^{t}) = \frac{\sum_{v \in POF^{t*}} \min_{v \in POF^{t}} \parallel v^{*} - v \parallel}{\mid POF^{t*} \mid} \tag{11}$$

where POF^{t*} is the projection of the uniformly distributed true POS at time window t and POF^{t} represents the approximated POF obtained by the algorithm, respectively. In the experiments, 1500 and 2500 points are uniformly sampled on the POF^{t*} of two-objective problems and three-objective problems, respectively, to compute the IGD metrics. A lower value of IGD implies that the algorithm has better optimization quality. To obtain a low value of IGD, the POF^{t} should be close enough to the true POF POF^{t*}.

The Mean Inverted Generational Distance (MIGD) [16], [17] is a variant of IGD which is modified to evaluate DMOEAs, it calculates the average of the IGD values in a certain number of time periods over a single run, given by:

$$\text{MIGD}(POF^{t*}, POF^{t}) = \frac{1}{T}\sum_{t \in T} \text{IGD}(POF^{t*}, POF^{t}) \tag{12}$$

where T is a set of discrete time periods.

TABLE II
MEAN AND STANDARD DEVIATIONS OF MIGD VALUES OBTAINED BY THE COMPARED ALGORITHMS ON DMOPS POSSESSING THREE OBJECTIVES

		MOEA/D Optimizer		*MOPSO Optimizer*		
Problems	(n_t,τ_t)	*Tr-MOEA/D*	*OP-MOEA/D*	*MDP-MOPSO*	*Tr-MOPSO*	*OP-MOPSO*
DF10	(5,10)	1.3621E-01 (8.1155E-03)≈	**1.2652E-01 (2.3681E-02)**	2.2542E-01 (4.1795E-03)≈	2.6226E-01 (1.5775E-02)+	**2.1934E-01 (1.6415E-02)**
	(2.5,10)	**1.6812E-01 (2.0096E-02)**≈	1.8162E-01 (1.5115E-02)	2.2940E-01 (6.0724E-03)≈	2.8869E-01 (1.2301E-02)+	**2.2467E-01 (2.3262E-02)**
	(1,10)	**2.1716E-01 (3.2408E-02)**≈	2.3831E-01 (3.3606E-02)	2.2670E-01 (7.3124E-03)+	2.9898E-01 (1.3314E-02)+	**1.9312E-01 (1.4250E-02)**
DF11	(5,10)	6.6853E-01 (5.4275E-03)+	**6.6316E-01 (1.3526E-02)**	7.1543E-01 (2.7151E-03)+	3.2339E+01 (4.2026E-02)+	**7.0797E-01 (1.7309E-02)**
	(2.5,10)	6.6218E-01 (2.1266E-03)+	**6.6064E-01 (5.3725E-03)**	7.0902E-01 (3.2705E-03)+	3.2274E+01 (2.6487E-02)+	**7.0017E-01 (6.3414E-03)**
	(1,10)	**5.7089E-01 (2.2866E-03)**≈	5.8153E-01 (2.2644E-02)	6.1020E-01 (1.9451E-02)+	3.1682E+01 (9.6272E-02)+	**5.7390E-01 (4.2007E-03)**
DF12	(5,10)	2.7948E-01 (4.1479E-02)+	**1.0551E-01 (9.9298E-03)**	6.0747E-01 (2.9697E-02)+	6.8831E-01 (5.3918E-02)+	**2.0201E-01 (1.1212E-02)**
	(2.5,10)	2.2647E-01 (2.5067E-02)+	**1.1508E-01 (8.7657E-03)**	5.7491E-01 (2.4269E-02)+	6.3474E-01 (2.7760E-02)+	**1.9379E-01 (8.2697E-03)**
	(1,10)	2.4811E-01 (1.8299E-02)+	**1.8025E-01 (1.1810E-02)**	5.3045E-01 (2.6535E-02)+	5.8177E-01 (5.2169E-02)+	**2.1375E-01 (1.1531E-02)**
DF13	(5,10)	3.3996E-01 (2.7335E-02)+	**2.7685E-01 (2.4950E-02)**	3.9910E-01 (2.4519E-02)+	1.1538E+00 (1.3014E-01)+	**1.9434E-01 (1.0502E-02)**
	(2.5,10)	3.4271E-01 (2.9593E-02)+	**2.6955E-01 (3.7015E-02)**	5.7352E-01 (5.2451E-02)+	1.1718E+00 (1.4145E-01)+	**2.1778E-01 (1.4630E-02)**
	(1,10)	3.6518E-01 (4.1525E-02)+	**2.9568E-01 (3.9705E-02)**	5.5972E-01 (4.0483E-02)+	1.1397E+00 (1.0005E-01)+	**2.6790E-01 (2.8651E-02)**
DF14	(5,10)	6.4360E-02 (1.9245E-03)≈	**6.2030E-02 (4.3103E-03)**	1.9178E-01 (2.0239E-02)+	1.7449E-01 (3.1672E-02)≈	**9.1416E-02 (3.1123E-02)**
	(2.5,10)	6.2861E-02 (2.6633E-03)+	**6.1762E-02 (5.7728E-03)**	2.7141E-01 (3.8948E-02)+	1.5267E-01 (2.9497E-02)+	**8.6230E-02 (1.2471E-02)**
	(1,10)	5.6179E-02 (6.7604E-03)≈	**5.1052E-02 (6.4459E-03)**	5.6948E-01 (1.3235E-01)+	9.7977E-02 (1.5858E-02)+	**4.9186E-02 (4.8722E-03)**

Superior performances are highlighted in bold. "+", "–", and "≈" denote proposed OP-DMOEA is statistically significant better, worse, and similar to the compared dynamic multi-objective method which shares the same multi-objective optimizer, respectively.

3) Compared Algorithms: To confirm the efficacy of the proposed prediction method for solving DMOPs, three state-of-the-art DMOEAs based on two different optimization solvers, are considered here as the baseline algorithms for comparison. In particular, the first one is *Tr-MOPSO* proposed by Jiang *et al.* in [17], while the second one integrates the same transfer learning framework as in [17] with the MOEA/D solver, named *Tr-MOEA/D*. The third baseline is a multidirectional POS prediction approach based on the multi-objective particle swarm optimization solver, called *MDP-MOPSO*, proposed by Rong *et al.* in [25]. For fair comparison, we integrate our proposed prediction in objective space with MOPSO (labeled as *OP-MOPSO*), and MOEA/D (labeled as *OP-MOEA/D*), and compare them with the baseline algorithm possessing the same evolutionary solver, respectively.

In the experiments, according to [30], for each studied DMOP, all the algorithms are supposed to stop after 30 environmental changes. The population size of all the compared algorithms is set as 100, number of variables is set as 10. Other configurations of evolutionary operators and parameters are kept consistent with the settings in [25] and [17] for MOPSO.

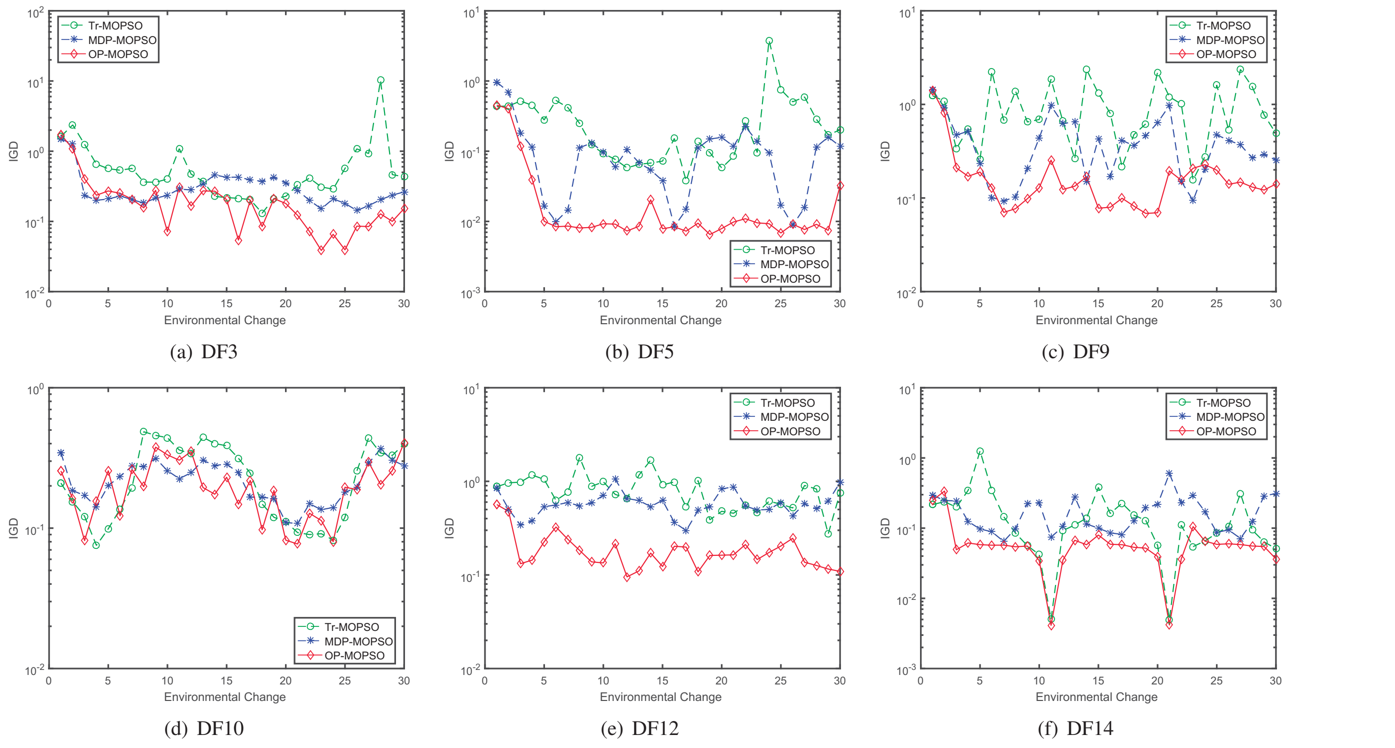
B. Results and Discussion

In this section, we analyse the comparison results first from the perspective of solution quality, which is followed by the discussion on the convergence speed of the compared algorithms after the dynamic occurs.

1) Solution Quality: Averaged MIGD values and the standard deviations obtained by *Tr-MOEA/D*, *OP-MOEA/D*, *MDP-MOPSO*, *Tr-MOPSO* and *OP-MOPSO* under the frameworks of MOEA/D and MOPSO, respectively, are presented in Table I and Table II. In the tables, over 20 independent runs on the two-objective and three-objective DMOPs, comparison results are summarized and superior performances are highlighted in bold. Further, the Wilcoxon rank sum test with 95% confidence level is conducted on the results. "+", "–", and "≈" denote proposed OP-DMOEA is statistically significant better, worse, and similar to the compared method which shares the same multi-objective optimizer, respectively.

As can be observed in the tables, the proposed *OP-MOPSO* performs best on all instances of DF3∼DF5, DF8, DF9 and five three-objective DMOP benchmarks, as well as the proposed *OP-MOEA/D* achieves superior performance on all instances of DF3, DF5, DF6, DF9 and DF12∼DF14 in the comparison with *Tr-MOEA/D*. The transfer learning based DMOEAs (i.e., *Tr-MOPSO* and *Tr-MOEA/D*) obtains better results on most cases of DF1 and DF2. DF2 has a simple dynamic on the PS and its PF remains stationary over time, but the switch of the position-related variable is a challenging dynamic [30], which makes *Tr-MOPSO* and *Tr-MOEA/D* more suitable to solve this kind of DMOP. Because of the special characteristics of DF6 and DF7, it is difficult to find good solutions within the given optimization budget [20]. Therefore the poor quality of historical solutions weakens the performance of these compared prediction methods. In summary, on totally 42 number of DMOP instances, *OP-MOPSO* and *OP-MOEA/D* obtained superior or competitive MIGD values over the compared DMOEAs with same optimizer, on 30 and 35 instances, respectively.

2) Convergence Speed: The IGD values of the tracking dynamics for six representative DMOP benchmarks obtained by three prediction-based DMOES based on the MOPSO solver are depicted in Fig. 1. In the figures, Y-axis denotes the averaged IGD values, while the X-axis gives the index of dynamic changes. It can be observed in Fig. 1 that on all the studied DMOP benchmarks, the proposed *OP-MOPSO*



(a) DF3 (b) DF5 (c) DF9

(d) DF10 (e) DF12 (f) DF14

Fig. 1. Convergence curves of the averaged IGD (over 20 runs) obtained by *Tr-MOPSO, MDP-MOPSO,* and *OP-MOPSO* on representative DMOPs, with $n_t = 5, \tau_t = 10$, (Y-axis: IGD value; X-axis: Index of dynamic change).

achieved superior performance in terms of tracking dynamic changes, compared to *Tr-MOPSO* and *MDP-MOPSO*.

The graphical view of the tracking ability shows that the proposed prediction method responds to changes more stably and recovers faster on most of the test inctances, thereby obtaining superior convergence performance when compared with the other approaches.

V. Conclusion

In this paper, we have proposed a new prediction method which builds the prediction model in Reproducing Kernel Hilbert Space, to track the moving of POF in objective space for solving DMOP. In particular, when a change is detected, the prediction model with a closed-form solution, is derived to generate the initial population for the evolutionary search in next time instance. To evaluate the performance of the proposed prediction in objective space, empirical studies have been conducted on IEEE CEC2018 DMOP benchmarks under three different dynamic configurations. The results obtained on two evolutionary optimizer confirmed the superior performance of the proposed method in terms of both solution quality and convergence speed.

For future work, we would like to apply the proposed prediction method to solve DMOP applications in real-world scenarios, such as dynamic optimization of recommendation system and dynamic path planning.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61876025, 61803054 and 61871272, the Venture & Innovation Support Program for Chongqing Overseas Returnees under Grants No. cx2018044 and cx2019020, the Fundamental Research Funds for the Central Universities (2019CDQYZD-H030), the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20190808173617147.

The authors would like to thank Dr. M. Jiang for providing the source code of their transfer learning based algorithm [17].

References

[1] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[2] C. A. C. Coello, “Evolutionary multi-objective optimization: a historical view of the field,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, Feb 2006.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] C. C. Coello and M. S. Lechuga, “MOPSO: A proposal for multiple objective particle swarm optimization,” in *IEEE Congress on Evolutionary Computation. CEC'02*, vol. 2. IEEE, 2002, pp. 1051–1056.

[5] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[6] L. Huang, I. H. Suh, and A. Abraham, “Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants,” *Information Sciences*, vol. 181, no. 11, pp. 2370–2391, 2011.

[7] D.-J. Wang, F. Liu, and Y. Jin, "A multi-objective evolutionary algorithm guided by directed search for dynamic scheduling," *Computers & Operations Research*, vol. 79, pp. 279–290, 2017.

[8] W. K. Mashwani and A. Salhi, "Multiobjective evolutionary algorithm based on multimethod with dynamic resources allocation," *Applied Soft Computing*, vol. 39, pp. 292–309, 2016.

[9] M. Mavrovouniotis, F. M. Müller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1743–1756, 2016.

[10] Y.-N. Guo, J. Cheng, S. Luo, D. Gong, and Y. Xue, "Robust dynamic multi-objective vehicle routing optimization method," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1891–1903, 2017.

[11] S. Bechikh, R. Datta, and A. Gupta, *Recent Advances in Evolutionary Multi-objective Optimization*. Springer International Publishing, 2017, vol. 20.

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

[13] K. Deb, S. Karthik *et al.*, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling," in *International conference on evolutionary multi-criterion optimization*. Springer, 2007, pp. 803–817.

[14] C. Li and S. Yang, "A general framework of multipopulation methods with clustering in undetectable dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 556–577, 2012.

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

[16] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via kalman filter prediction," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2016.

[17] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer learning based dynamic multiobjective optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–1, 2016.

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

[19] S. Sahmoud and H. R. Topcuoglu, "A memory-based NSGA-II algorithm for dynamic multi-objective optimization problems," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 296–310.

[20] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, "Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2019.

[21] D. W. Iason Hatzakis, "Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach," in *Conference on Genetic & Evolutionary Computation*, vol. 4. ACM, 2006, pp. 1201–1208.

[22] W. T. Koo, C. K. Goh, and K. C. Tan, "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment," *Memetic Computing*, vol. 2, no. 2, pp. 87–110, 2010.

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

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

[25] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, "Multidirectional prediction approach for dynamic multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3362–3374, 2018.

[26] S. Yang and X. Yao, *Evolutionary Computation for Dynamic Optimization Problems*. Springer International Publishing, 2013.

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

[28] L. Feng, Y. S. Ong, S. Jiang, and A. Gupta, "Autoencoding evolutionary search with learning across heterogeneous problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 760–772, 2017.

[29] X. He and P. Niyogi, "Locality preserving projections," in *Advances in neural information processing systems*, vol. 16, 2004, pp. 153–160.

[30] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser, and N. Krasnogor, "Benchmark problems for cec2018 competition on dynamic multiobjective optimisation," *Tech. Rep.*, 2018.

[31] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. D. Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.