




An Online Machine Learning-Based Prediction Strategy for Dynamic Evolutionary Multi-objective Optimization

Min Liu^{1,2} , Diankun Chen¹, Qiongbing Zhang¹, and Lei Jiang^{1,2}

¹ School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 4111201, China

liumin_xt@163.com, 2059426443@qq.com, mrtly2@whu.edu.cn, jleihn@hotmail.com

² Key Laboratory of Knowledge Processing and Networked Manufacturing, College of Hunan Province, Xiangtan 4111201, China

Abstract. Due to the impact of environmental changes, dynamic evolutionary multi-objective optimization algorithms need to track the time-varying Pareto optimal solution set of dynamic multi-objective optimization problems (DMOPs) as soon as possible by effectively mining historical data. Since online machine learning can help algorithms dynamically adapt to new patterns in the data in machine learning community, this paper introduces Passive-Aggressive Regression (PAR, a common online learning technology) into dynamic evolutionary multi-objective optimization research area. Specifically, a PAR-based prediction strategy is proposed to predict the new Pareto optimal solution set of the next environment. Furthermore, we integrate the proposed prediction strategy into the multi-objective evolutionary algorithm based on decomposition with a differential evolution operator (MOEA/D-DE) to handle DMOPs. Finally, the proposed prediction strategy is compared with three state-of-the-art prediction strategies under the same dynamic MOEA/D-DE framework on CEC2018 dynamic optimization competition problems. The experimental results indicate that the PAR-based prediction strategy is promising for dealing with DMOPs.

Keywords: Evolutionary multi-objective optimization · Dynamic environment · Prediction strategy · Online machine learning

1 Introduction

In the real world, there exist a lot of dynamic multi-objective optimization problems (DMOPs) which usually involve several conflicting objective functions and may be subject to a number of constraints. Furthermore, the objective functions, constraints and/or relative parameters of DMOPs may change over time [1].

In this work, we consider the following continuous DMOPs:

$$\begin{cases} \min_{\mathbf{v} \in \Omega} F(\mathbf{v}, t) = (f_1(\mathbf{v}, t), f_2(\mathbf{v}, t), \dots, f_m(\mathbf{v}, t))^T \\ \text{s.t. } h_i(\mathbf{v}, t) \leq 0, i = 1, 2, \dots, p; \quad l_j(\mathbf{v}, t) = 0, j = 1, 2, \dots, r \end{cases} \quad (1)$$

where \mathbf{v} is the decision vector defined in the decision space Ω , $F(\mathbf{v}, t)$ is the objective vector that calculates a numerical value for each objective of the solution \mathbf{v} at time t , m is the number of objectives. The functions h_i and l_j are the inequality and equality constraints which change over t , respectively. As a result, the Pareto optimal solution set (POS) in the decision space and/or Pareto optimal front (POF) in the objective space of DMOPs may also change over time, which imposes a big challenge to evolutionary multi-objective optimization (EMO) researches.

Over the past decade or so, there are increasing interests in designing dynamic evolutionary multi-objective optimization (DEMO) algorithms to handle DMOPs. The task of a DEMO is to trace the movement of the POS and/or POF with reasonable computational costs. How to take action to respond to a new change is the key issue of DEMO, since an effective change reaction method can help the DEMO algorithm adapt to the new environment quickly. A lot of change reaction methods have been proposed to handle changes. These methods include diversity introduction after a change occurs [2, 3] or maintain diversity throughout the run [4], multi-population approaches [5], memory schemes [6, 7], and prediction strategies [8–14].

In recent years, prediction strategies have gained much attention. This kind of methods predicts POS of the new environment in advance by exploiting different machine learning techniques. According to the amount of historical information used for prediction, the prediction strategies can be roughly categorized into two types: local information-based prediction (LIP) and time series-based prediction (TSP). LIP approaches usually only use information obtained from several recent environments to make predictions, so that the history data has not been fully mined. Some representative LIP methods are simple linear model [9], Kalman filter [11], differential model [12], and transfer learning [13]. In contrast, TSP approaches make use of much more historical environmental information than LIP approaches. TSP approaches first collect approximate optimal solutions from past environments as training data, and then feed them to some machine learning model to predict the location of the new optimal solutions. Therefore, TSP approaches generally have higher training costs than LIP approaches. Autoregressive model [8, 10] and support vector regression [14] are some well-known models used in TSP approaches.

The Motivation of This Work. Dealing with a dynamic and uncertain environment is not a unique challenge to evolutionary optimization. There are also active research activities within online machine learning community that try to tackle similar challenges from changing data streams [15]. Online learning is used to update the prediction model for future data at each time step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once [16]. Online learning is also used in situations

where it is necessary for the algorithm to dynamically adapt to new patterns in the data [17]. In addition, online learning algorithms are typically easy to implement and generally have low computational cost [18], which is particularly suitable for dynamic optimization. Inspired by the aforementioned advantages of online learning, we introduce online learning into DEMO in order to help dynamic optimization algorithms better adapt to changing environments.

The Contribution of This Work. We first introduce Passive-Aggressive Regression (PAR, a common online learning technique) into DEMO, and then propose a PAR-based prediction strategy (PARS) to react the new change of DMOPs. Furthermore, we incorporate this prediction strategy into MOEA/D-DE [19] to deal with DMOPs. Finally, in order to fairly evaluate the performance of the proposed prediction strategy, it is compared with three state-of-the-art prediction methods under the same dynamic MOEA/D-DE framework.

The rest of this paper is organized as follows. Section 2 provides a brief survey on the prediction strategies for DEMO, and introduces the basic knowledge of PAR. In Sect. 3, PARS is proposed, and then it is incorporated into MOEA/D-DE to handle DMOPs. In the following section, some experimental results are reported to show the effectiveness of our proposed strategy. The final section concludes this paper.

2 Related Works

2.1 Prediction Strategies for DEMO Algorithm

Local Information-Based Prediction Strategies: Zhou et al. [9] proposed a prediction strategy (PRE) in 2007. In PRE, a simple linear prediction model with Gaussian noise is used to reinitialize population in new environment by utilizing only optimal individuals of last two environments. The idea of this simple linear prediction model was employed by other algorithms which focused predicted optimal individuals on some special points [7, 20] or multiple directions [21]. Instead of making predictions based on data from the previous two environments, Cao et al. [12] proposed a differential model for predicting the movement of the centroid by its locations in three former environments. Muruganantham et al. [11] proposed a Kalman filter (KF) prediction model to guide the search toward a new POS. This KF model was essentially implemented as first- or second-order linear prediction model [14]. Recently, Jiang et al. [13] only utilized the approximate optimal solutions under two sequential environments to construct a prediction model by adopting transfer learning.

Time Series-Based Prediction Strategies: Zhou et al. [10] proposed a population-based prediction strategy (PPS). The movement of centers of the population is learnt by a univariate autoregressive model. The predicted center point and estimated manifold are utilized to generate a new population for the next environment. Cao et al. [14] proposed a support vector regression (SVR) predictor for better solving DMOPs with nonlinear correlation. Whenever there

is a new change taken place, the historical approximate optimal solutions in time series are first used to train a group of SVR models. Then the POS of the next environment can be predicted by using the trained SVR models.

2.2 Online Passive-Aggressive Regression

In 2006, Crammer et al. [18] proposed a Passive-Aggressive Regression (PAR) which has become a common online machine learning method. To deal with noises of the samples, two PAR variants (i.e., PAR-I and PAR-II) were also proposed in [18]. In this paper, we only focus on PAR-II just for simplicity. On each time step, the PAR-II algorithm receives an instance $\mathbf{x}^t \in R^n$ and predicts a target value $\hat{y}^t \in R$ using its regression function, that is, $\hat{y}^t = \mathbf{w}^t \cdot \mathbf{x}^t$ where \mathbf{w}^t is the incrementally learned vector. After prediction, the algorithm obtains the true target value y^t and suffers an instantaneous loss. PAR-II uses the ε -insensitive hinge loss function:

$$\ell_\varepsilon(\mathbf{w}^t; (\mathbf{x}^t, y^t)) = \begin{cases} 0, & |\mathbf{w}^t \cdot \mathbf{x}^t - y^t| \leq \varepsilon \\ |\mathbf{w}^t \cdot \mathbf{x}^t - y^t| - \varepsilon, & \text{otherwise} \end{cases} \quad (2)$$

where ε is a positive parameter which controls the sensitivity to prediction errors. At the end of each time step, the algorithm uses \mathbf{w}^t and the example (\mathbf{x}^t, y^t) to generate a new weight vector \mathbf{w}^{t+1} , which will be utilized to continue making a prediction on the next time step. Firstly, \mathbf{w}^1 is initialized to $(0, \dots, 0)$. On each time step, the PAR-II algorithm updates the new weight vector to be,

$$\mathbf{w}^{t+1} = \arg \min_{\mathbf{w} \in R^n} \frac{1}{2} \|\mathbf{w} - \mathbf{w}^t\|^2 + C\xi^2 \quad \text{s.t.} \quad \ell_\varepsilon(\mathbf{w}; (\mathbf{x}^t, y^t)) \leq \xi \quad (3)$$

where ξ is a non-negative slack variable and C is a positive parameter which controls the influence of the slack term on the objective function. Specifically, larger values of C imply a more aggressive update step and therefore C is referred to as the aggressiveness parameter of the algorithm. Using the shorthand $\ell^t = \ell_\varepsilon(\mathbf{w}^t; (\mathbf{x}^t, y^t))$, the update given in Eq. (3) has a closed form solution as follows,

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \text{sign}(y^t - \hat{y}^t) \tau^t \mathbf{x}^t \quad \text{where} \quad \tau^t = \frac{\ell^t}{\|\mathbf{x}^t\|^2 + \frac{1}{2C}} \quad (4)$$

This update keeps a balance between the amount of progress made on each time step and the amount of information retained from previous time steps [18]. On one hand, this update requires \mathbf{w}^{t+1} to correctly predict the current example with a sufficiently high accuracy and thus progress is made. On the other hand, \mathbf{w}^{t+1} must be as close to \mathbf{w}^t as possible in order to retain the information learned from previous time steps. After given the loss function Eq. (2) and the update rule Eq. (4), the pseudo-code of PAR-II is presented in Algorithm 1.

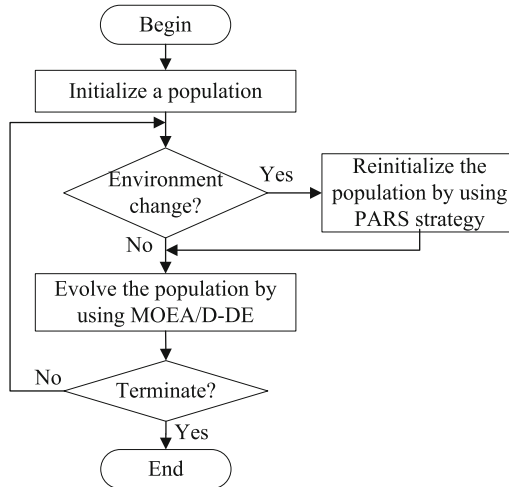
Algorithm 1. Passive-Aggressive Regression II (PAR-II) [18]

Input: aggressiveness parameter $C > 0$;

- 1: Initialize $\mathbf{w}^1 = (0, 0, \dots, 0)$;
- 2: **for** $t=1, 2, \dots$ **do**
- 3: receive instance \mathbf{x}^t ;
- 4: receive label y^t ;
- 5: predict: $\hat{y}^t = \mathbf{w}^t \cdot \mathbf{x}^t$;
- 6: suffer loss: $\ell^t = \ell_\varepsilon(\mathbf{w}^t; (\mathbf{x}^t, y^t))$; /*according to Eq. 2*/
- 7: update: /* according to Eq. 4*/
- 8: $\tau^t = \ell^t / (\|\mathbf{x}^t\|^2 + 1/(2C))$;
- 9: $\mathbf{w}^{t+1} = \mathbf{w}^t + \text{sign}(y^t - \hat{y}^t)\tau^t \mathbf{x}^t$
- 10: **end for**

3 Proposed PAR-Based Prediction Strategy

Whenever there is a new change taken place, DEMO algorithm need make some operation to response the new change so as to adapt to new environment as quickly as possible. In the following, we will propose a new prediction strategy to adapt to the new change by using PAR-II. Algorithm 2 presents our PAR-based strategy (PARS) for predicting the new POS of the next environment. We assume that there are N individuals in the population and each individual has d decision variables. Because we perform regression prediction on each dimension variable separately, there are a total of $N * d$ PAR-II regression models in this algorithm. In addition, we assume that the dimension of the weight vector \mathbf{w} of PAR-II is q . Firstly line 3 is to generate a time series s from the stored populations. Then, M samples are gotten from s in line 4. In order to reduce

**Fig. 1.** MOEA/D-PARS algorithm framework

Algorithm 2. Passive-Aggressive Regression-based Strategy (PARS)

Input: the stored populations of the preceding time steps P^1, P^2, \dots, P^t , where $P^k = \{v_{ij}^k | i = 1, 2, \dots, N; j = 1, 2, \dots, d; k = 1, 2, \dots, t\}$

Output: a predicted population of the next time step P^{t+1} .

- 1: **for** $i=1, 2, \dots, N$ **do**
- 2: **for** $j=1, 2, \dots, d$ **do**
- 3: Generate a time series $s = (v_{ij}^1, v_{ij}^2, \dots, v_{ij}^t)$ from the stored populations;
- 4: Get M training samples from s ;
- 5: Initialize weight vector of PAR-II model: $w^1 = (0, 0, \dots, 0)$;
- 6: **for** $k=1, 2, \dots, M$ **do** /*Train the model*/
- 7: receive instance $x^k = (v_{ij}^{t-q-k+1}, \dots, v_{ij}^{t-k})$ from the time series s ;
- 8: receive label $y^k = v_{ij}^{t-k+1}$ from time series s ;
- 9: predict: $\hat{y}^k = w^k \cdot x^k$;
- 10: suffer loss: $\ell^k = \ell_\epsilon(w^k; (x^k, y^k))$; /*according to Eq. 2*/
- 11: update: /* according to Eq. 4*/
- 12: $\tau^k = \ell^k / (\|x^k\|^2 + 1/(2C))$;
- 13: $w^{k+1} = w^k + \text{sign}(y^k - \hat{y}^k)\tau^k x^k$
- 14: **end for** /*end of the training procedure*/
- 15: set $w^{t+1} = w^{M+1}$;
- 16: set test instance $x^{t+1} = (v_{ij}^{t-q+1}, \dots, v_{ij}^t)$;
- 17: predict: $v_{ij}^{t+1} = w^{t+1} \cdot x^{t+1}$, and repair v_{ij}^{t+1} if it is out of the boundary [10];
- 18: **end for**
- 19: **end for**
- 20: **return** $P^{t+1} = (v_1^{t+1}, \dots, v_N^{t+1})$.

the training time, the size of training set is limited to no more than 20, so the size M is set to $\min(20, t - q)$. Referring to Algorithm 1, line 5 to 15 is used to implement a PAR-II. Line 17 predicts one component value of a Pareto optimal solution on time step $t+1$ and checks whether it is out of the boundary of the decision space. Since Algorithm 2 has three loops, its overall time complexity is $O(N * d * M)$.

In Fig. 1, we further integrate PARS into MOEA/D-DE to generate a PARS assisted dynamic MOEA/D-DE algorithm (termed MOEA/D-PARS) which is mainly composed of an environment change detection operator, PARS prediction strategy and a MOEA/D-DE optimization [19]. A change detection operator suggested in [2] is adopted to detect whether a new change has occurred. 10% of individuals in the population are selected randomly as detectors and re-evaluated in every generation. If any detector's objective values are different from its previous ones, then we assume a change has taken place. If a new environmental change is detected, PARS is called to respond to the change. Later on, the population is evolved for a generation by using MOEA/D-DE. Finally, if the stop condition is not met, the algorithm continues tracking the next dynamic change. Considering the time complexity of MOEA/D-PARS, we assume that the environmental change frequency is τ_T , that is, there are τ_T generations of evolutionary multi-objective optimization in an environmental change. The time complexities of the

change detect operator, PARS, and MOEA/D-DE are $O(mN)$, $O(N*d*M)$, and $O(mNT)$ [19] respectively, where T is the number of weight vectors, and m is the number of objective functions. Therefore, the time complexity of MOEA/D-PARS during an environmental change is $O(\tau_T mNT)$.

4 Computational Experiments

4.1 Experimental Settings

Benchmark Problems. Fourteen different benchmark problems for CEC2018 competition on dynamic multi-objective optimization [22] are chosen here to compare the performance of DEMO algorithms.

Compared Prediction Strategies and DEMO Algorithms. Three well-known prediction strategies, i.e., PRE [9], PPS [10], and KF [11], are chosen here to validate the performance of the proposed PARS. For fairly comparison, we choose MOEA/D-DE as their basic multi-objective optimizer. In addition to the proposed MOEA/D-PARS algorithm, we integrate the aforementioned PRE, PPS, and KF into MOEA/D-DE, and then we get other three compared dynamic MOEA/D-DE algorithms which are termed MOEA/D-PRE, MOEA/D-PPS, and MOEA/D-KF respectively. All the algorithms are implemented by using Python programming.

Performance Metrics. Mean Inverted Generational Distance (MIGD) [11, 14] is selected as performance metrics. It provides a quantitative measurement for both the proximity and diversity goals of DEMO algorithms. A lower value of MIGD implies that the algorithm has better optimization performance. In addition, the IGD(t) [23] is used to illustrate the evolutionary tracking curves of the compared DEMO algorithms.

Parameter Setting. The parameter settings of all the DEMO algorithms tested in this paper were almost the same as those of their original papers. Some key parameters in these algorithms were set as follows:

- 1) Population size (N) and the number of decision variables (d): They were set to be 100 and 10 respectively for all benchmark problems.
- 2) Common control parameters in MOEA/D-DE: $CR=0.5$, $F=0.5$ in the differential evolution operator. Distribution $\eta_m=20$, mutation probability $p_m=1/d$ in the polynomial mutation operator. In addition, neighborhood size $T=20$ and neighborhood selection probability $\delta=0.8$.
- 3) Parameters setting for PARS strategy: $q=4$, $C=1$, $\varepsilon=0.05$.

Table 1. Mean and standard deviation values of MIGD obtained by four algorithms on the DF1–DF7 problems

Problem	(τ_T, n_T)	MOEA/D-PRE	MOEA/D-PPS	MOEA/D-KF	MOEA/D-PARS
DF1	(5, 5)	4.53e-2(3.10e-3)	5.86e-2(1.09e-2)	5.48e-2(8.83e-3)	4.17e-2(9.36e-3)
	(5, 10)	1.88e-2(1.54e-4)	3.52e-2(7.16e-3)	2.10e-2(2.00e-3)	2.75e-2(2.20e-3)
	(10, 5)	1.42e-2(3.26e-3)	1.33e-2(2.08e-3)	1.33e-2(1.90e-3)	1.14e-2(4.11e-4)
	(10, 10)	8.40e-3(2.87e-4)	1.24e-2(2.41e-3)	8.19e-3(3.31e-4)	1.00e-2(4.23e-4)
DF2	(5, 5)	1.07e-1(5.37e-3)	6.81e-2(5.86e-3)	1.40e-1(9.48e-3)	4.73e-2(3.69e-3)
	(5, 10)	6.90e-2(5.62e-3)	5.38e-2(6.50e-3)	7.84e-2(5.06e-3)	4.30e-2(3.88e-3)
	(10, 5)	2.08e-2(1.98e-3)	1.76e-2(9.86e-4)	2.50e-2(1.87e-3)	1.29e-2(8.66e-4)
	(10, 10)	1.52e-2(1.39e-3)	1.56e-2(1.21e-3)	1.66e-2(2.64e-3)	1.26e-2(9.10e-4)
DF3	(5, 5)	3.38e-1(3.12e-3)	4.74e-1(1.32e-2)	3.22e-1(2.49e-2)	2.89e-1(3.36e-2)
	(5, 10)	2.34e-1(2.13e-2)	4.79e-1(8.50e-3)	3.02e-1(2.99e-2)	3.31e-1(2.92e-2)
	(10, 5)	2.16e-1(2.40e-2)	3.92e-1(1.68e-2)	2.43e-1(2.64e-2)	2.26e-1(4.55e-2)
	(10, 10)	1.54e-1(2.55e-3)	3.91e-1(2.53e-2)	1.99e-1(2.48e-2)	2.40e-1(1.67e-2)
DF4	(5, 5)	7.84e-2(1.07e-3)	9.20e-2(1.95e-3)	7.99e-2(2.53e-3)	7.64e-2(5.35e-3)
	(5, 10)	8.11e-2(1.97e-3)	9.81e-2(3.16e-3)	8.23e-2(2.12e-3)	8.39e-2(4.81e-3)
	(10, 5)	6.36e-2(3.97e-4)	6.39e-2(5.47e-4)	6.42e-2(7.85e-3)	6.25e-2(8.87e-4)
	(10, 10)	7.17e-2(6.42e-4)	7.20e-2(1.07e-3)	7.23e-2(6.64e-4)	7.19e-2(9.22e-4)
DF5	(5, 5)	1.56e-2(4.27e-4)	2.15e-2(3.98e-3)	1.26e-2(2.31e-4)	1.26e-2(7.38e-4)
	(5, 10)	1.26e-2(3.26e-4)	1.96e-2(1.80e-3)	1.14e-2(2.31e-4)	1.82e-2(4.49e-4)
	(10, 5)	8.18e-3(9.03e-5)	8.77e-3(8.97e-5)	7.37e-3(3.91e-5)	6.63e-3(8.36e-5)
	(10, 10)	7.27e-3(4.95e-5)	8.75e-3(1.26e-4)	6.69e-3(5.15e-5)	8.74e-3(1.22e-4)
DF6	(5, 5)	2.18e+0(4.60e-1)	1.38e+0(3.52e-1)	1.73e+0(6.02e-1)	9.86e-1(3.37e-1)
	(5, 10)	2.01e+0(7.23e-1)	1.66e+0(4.47e-1)	2.81e+0(1.24e+0)	8.42e-1(3.31e-1)
	(10, 5)	1.05e+0(4.71e-1)	9.14e-1(2.43e-1)	1.43e+0(2.92e-1)	1.04e+0(5.01e-1)
	(10, 10)	1.25e+0(5.92e-1)	1.13e+0(4.01e-1)	2.89e+0(1.27e+0)	5.99e-1(2.23e-1)
DF7	(5, 5)	1.53e+0(1.09e-1)	1.81e+0(2.35e-2)	1.43e+0(5.00e-2)	1.51e+0(1.51e-1)
	(5, 10)	5.73e-1(1.37e-1)	8.58e-1(4.28e-2)	5.64e-1(1.56e-1)	7.41e-1(7.56e-2)
	(10, 5)	1.44e+0(1.19e-1)	1.69e+0(4.19e-2)	1.28e+0(3.12e-2)	1.71e+0(2.94e-2)
	(10, 10)	4.78e-1(1.19e-1)	8.19e-1(3.24e-2)	2.91e-1(9.20e-2)	6.39e-1(1.18e-1)

4) Number of runs and stopping condition: For each tested algorithm, it was executed 20 runs on each test instance, and the experimental results were recorded to obtain the statistical information. In each run, each algorithm stopped at pre-specified 100 environmental changes. During each environmental change, each algorithm has the same number of function evaluations.

4.2 Experimental Results

The experiments were conducted with different combinations of change severity levels (n_T) and frequencies (τ_T) in order to study the impact of dynamics in changing environments. They were set to 5 and 10, respectively. Comparison results of four DEMO algorithms (i.e., MOEA/D-PRE, MOEA/D-PPS, MOEA/D-KF, and MOEA/D-PARS) are reported to study the influence of different prediction strategies on DEMO algorithms. Table 1 and Table 2 detail MIGD values of these algorithms on DF benchmark problems. These two tables show that MOEA/D-PARS obtains competitive results on the majority of the

Table 2. Mean and standard deviation values of MIGD obtained by four algorithms on the DF8–DF14 problems

Problem	(τ_T, n_T)	MOEA/D-PRE	MOEA/D-PPS	MOEA/D-KF	MOEA/D-PARS
DF8	(5, 5)	8.70e-2(2.55e-3)	1.04e-1(4.42e-3)	8.66e-2(7.66e-4)	8.45e-2(1.43e-3)
	(5, 10)	8.31e-2(2.47e-3)	1.25e-1(2.75e-2)	9.33e-2(5.41e-3)	8.55e-2(8.97e-3)
	(10, 5)	7.65e-2(1.06e-3)	8.72e-2(1.51e-3)	7.55e-2(9.02e-4)	7.90e-2(4.68e-4)
	(10, 10)	7.21e-2(1.35e-3)	9.31e-2(1.34e-2)	7.23e-2(1.47e-3)	7.77e-2(1.18e-1)
DF9	(5, 5)	2.75e-1(9.68e-2)	2.00e-1(1.90e-2)	4.56e-1(4.22e-2)	1.41e-1(1.13e-2)
	(5, 10)	1.17e-1(1.81e-2)	1.81e-1(1.93e-2)	1.65e-1(1.68e-2)	1.51e-1(1.61e-2)
	(10, 5)	1.23e-2(2.62e-2)	1.04e-1(7.33e-3)	2.16e-1(2.95e-2)	7.96e-2(8.62e-3)
	(10, 10)	7.83e-2(1.12e-2)	9.50e-2(1.14e-2)	7.64e-2(6.40e-3)	7.83e-2(1.46e-2)
DF10	(5, 5)	1.74e-1(3.16e-3)	2.35e-1(1.05e-2)	2.18e-1(1.33e-2)	1.50e-1(3.78e-3)
	(5, 10)	1.75e-1(1.16e-2)	2.37e-1(9.47e-3)	1.99e-1(1.48e-2)	1.49e-1(5.59e-3)
	(10, 5)	1.39e-1(1.86e-3)	1.75e-1(2.28e-2)	1.53e-1(1.27e-2)	1.31e-1(1.79e-3)
	(10, 10)	1.40e-1(2.58e-3)	1.65e-1(4.46e-3)	1.45e-1(3.35e-3)	1.30e-1(2.40e-3)
DF11	(5, 5)	7.24e-2(1.12e-4)	9.50e-2(6.58e-3)	7.22e-2(5.28e-4)	6.51e-2(6.98e-4)
	(5, 10)	6.83e-2(5.66e-4)	1.01e-1(7.12e-3)	6.92e-2(9.85e-4)	6.47e-2(7.43e-4)
	(10, 5)	5.80e-2(2.88e-4)	6.66e-2(2.75e-3)	5.84e-2(1.58e-4)	5.55e-2(3.98e-4)
	(10, 10)	5.69e-2(4.11e-4)	6.90e-2(7.34e-3)	5.69e-2(2.75e-4)	5.52e-2(2.02e-4)
DF12	(5, 5)	2.92e-1(1.15e-2)	3.91e-1(1.02e-2)	2.94e-1(2.34e-2)	3.00e-1(1.14e-2)
	(5, 10)	2.91e-1(1.01e-2)	3.88e-1(1.70e-2)	2.88e-1(1.87e-2)	2.81e-1(6.32e-3)
	(10, 5)	2.58e-1(1.15e-2)	3.05e-1(1.14e-2)	2.80e-1(1.62e-2)	2.74e-1(5.61e-3)
	(10, 10)	2.61e-1(1.15e-2)	3.07e-1(5.25e-3)	2.68e-1(1.43e-2)	2.59e-1(6.21e-3)
DF13	(5, 5)	2.25e-1(2.17e-4)	2.31e-1(6.13e-3)	2.23e-1(1.97e-3)	2.25e-1(2.84e-3)
	(5, 10)	2.22e-1(1.48e-3)	2.41e-1(2.63e-2)	2.18e-1(2.67e-3)	2.28e-1(1.57e-3)
	(10, 5)	2.26e-1(1.92e-3)	2.23e-1(2.33e-3)	2.27e-1(1.64e-3)	2.30e-1(2.35e-3)
	(10, 10)	2.23e-1(2.12e-3)	2.20e-1(1.90e-3)	2.24e-1(2.66e-3)	2.24e-1(1.96e-3)
DF14	(5, 5)	6.68e-2(1.79e-3)	7.71e-2(8.41e-3)	6.28e-2(1.09e-3)	6.86e-2(6.23e-4)
	(5, 10)	6.42e-2(4.86e-4)	7.84e-2(1.09e-2)	6.38e-2(1.19e-3)	6.86e-2(6.23e-4)
	(10, 5)	5.67e-2(2.27e-4)	6.01e-2(2.64e-3)	5.56e-2(6.18e-4)	5.58e-2(1.02e-3)
	(10, 10)	5.70e-2(1.77e-4)	6.01e-2(3.66e-4)	5.64e-2(2.35e-4)	5.85e-2(1.97e-4)

DF instances, which implies that MOEA/D-PARS has good performance regarding convergence and distribution. Specifically, MOEA/D-PARS obtains the best results on DF2, DF10, and DF11 instances, whereas MOEA/D-KF performs the best on DF5, DF7 and DF14 instances. MOEA/D-PARS also achieves the best results on DF1, DF3, DF4, DF6, and DF9 under some certain combinations of severity level and frequency. Besides, MOEA/D-PRE achieves the best results in a few cases. In most cases, the results obtained by MOEA/D-PPS are relatively poor.

Apart from presenting the data in the above tables, we also provide the average tracking performance of these algorithms over 20 runs in Fig. 2, where the average IGD values are plotted versus the time. For the sake of limitation of paper length, only results of four problems with $\tau_T = 10$ and $n_T = 5$ are presented here. It is clear that the tracking performance of MOEA/D-PARS is better than the other three algorithms on DF2 and DF10 problems. On DF4 problem, the tracking performance of the four algorithms is very similar. On DF6 problem, although MOEA/D-PARS keeps fluctuating in IGD values in the

first 50 time steps like the other three algorithms, it clearly outperformed the other competitors in the remaining time.

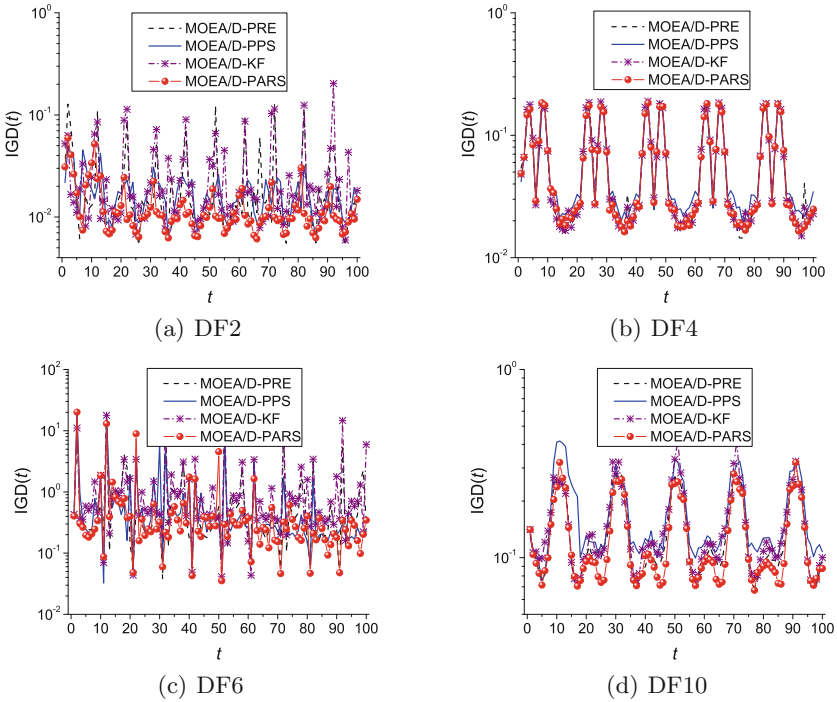


Fig. 2. Evolutionary average $IGD(t)$ curves obtained by four algorithms over 20 runs on the DF2, DF4, DF6, and DF10 problems with $\tau_T = 10$ and $n_T = 5$

5 Conclusions

Both evolutionary dynamic optimization and online machine learning communities face the challenge of dynamic and uncertain environment change problems. Hence, there may be an excellent opportunity for cross fertilization between them [15]. This paper first introduces PAR, a common online learning technique, into DEMO, and then proposes a PAR-based prediction strategy (PARS) to react the new environmental change of DMOPs. Furthermore, the proposed prediction strategy is integrated into MOEA/D-DE to deal with DMOPs. Finally, the proposed PARS is compared with three state-of-the-art prediction strategies under the same dynamic MOEA/D-DE framework. The experimental results show that MOEA/D-PARS is promising for handling DMOPs, which demonstrates that PARS is a competitive prediction strategy for DEMO.

The work presented here is preliminary, and there are some possible directions for future work. We would like to investigate the influence of parameter settings of PAR-II and other PAR variants on algorithm performance. In addition, we will try to design other prediction strategy based on other kind of online machine learning techniques for better dealing with DMOPs.

Acknowledgment. This work was supported by China Scholarship Council (Grant No. 201609480012) and a Project Supported by Scientific Research Fund of Hunan Provincial Education Department(Grant Nos. 18K060, 18C0331, 18B199).

References

1. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Trans. Evol. Comput.* **8**(5), 425–442 (2004)
2. Deb, K., Rao N., U.B., Karthik, S.: Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 803–817. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70928-2_60
3. Liu, M., Zheng, J.H., Wang, J.N., Liu, Y.Z., Jiang, L.: An adaptive diversity introduction method for dynamic evolutionary multiobjective optimization. In: Proceedings of 2014 IEEE Congress on Evolutionary Computation, CEC2014, Beijing, China, pp. 3160–3167. IEEE (2014)
4. Ruan, G., Yu, G., Zheng, J.H., Zou, J., Yang, S.X.: The effect of diversity maintenance on prediction in dynamic multi-objective optimization. *Appl. Soft Comput.* **58**, 631–647 (2017)
5. Goh, C.K., Tan, K.C.: A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Trans. Evol. Comput.* **13**(1), 103–127 (2009)
6. Liu, M., Zeng, W.H.: Memory enhanced dynamic multi-objective evolutionary algorithm based on decomposition. *Ruan Jian Xue Bao/J. Softw.* **24**(7), 1571–1588 (2013)
7. Peng, Z., Zheng, J., Zou, J., Liu, M.: Novel prediction and memory strategies for dynamic multiobjective optimization. *Soft Comput.* **19**(9), 2633–2653 (2014). <https://doi.org/10.1007/s00500-014-1433-3>
8. Hatzakis, I., Wallace, D.: Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2006, Seattle, USA, pp. 1201–1208. ACM (2006)
9. Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., Tsang, E.: Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 832–846. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70928-2_62
10. Zhou, A.M., Jin, Y.C., Zhang, Q.F.: A population prediction strategy for evolutionary dynamic multiobjective optimization. *IEEE Trans. Cybern.* **44**(1), 40–53 (2014)

11. Muruganantham, A., Tan, K.C., Vadakkepat, P.: Evolutionary dynamic multiobjective optimization via Kalman filter prediction. *IEEE Trans. Cybern.* **46**(12), 2862–2873 (2016)
12. Cao, L.L., Xu, L., Goodman, E.D., Li, H.: Decomposition-based evolutionary dynamic multiobjective optimization using a difference model. *Appl. Soft Comput.* **76**, 473–490 (2019)
13. Jiang, M., Huang, Z.Q., Qiu, L.M., Huang, W.Z., Yan, G.G.: Transfer learning-based dynamic multiobjective optimization algorithms. *IEEE Trans. Evol. Comput.* **22**(4), 501–514 (2018)
14. Cao, L.L., Xu, L., Goodman, E.D., Bao, C., Zhu, S.: Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor. *IEEE Trans. Evol. Comput.* **24**(2), 305–319 (2020)
15. Yao, X.: Challenges and opportunities in dynamic optimisation. In: *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, Amsterdam, The Netherlands, pp. 1761–1762. ACM (2013)
16. Sun, J.Y., Zhang, H., Zhou, A.M., Zhang, Q.F.: Learning from a stream of non-stationary and dependent data in multiobjective evolutionary optimization. *IEEE Trans. Evol. Comput.* **23**(4), 541–555 (2019)
17. Minku, L.L., Xin, Y.: DDD: a new ensemble approach for dealing with concept drift. *IEEE Trans. Knowl. Data Eng.* **24**(4), 619–633 (2012)
18. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. *J. Mach. Learn. Res.* **7**, 551–585 (2006)
19. Li, H., Zhang, Q.F.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **13**(2), 284–302 (2009)
20. Zou, J., Li, Q.Y., Yang, S.X., Bai, H., Zheng, J.H.: A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization. *Appl. Soft Comput.* **61**, 806–818 (2017)
21. Rong, M., Gong, D., Zhang, Y., Jin, Y.C., Pedrycz, W.: Multidirectional prediction approach for dynamic multiobjective optimization problems. *IEEE Trans. Cybern.* **49**(9), 3362–3374 (2019)
22. Jiang, S., Yang, S., Yao, X., Tan, K.C., Kaiser, M., Krasnogor, N.: Benchmark problems for CEC 2018 competition on dynamic multiobjective optimisation. In: *CEC 2018 Competition* (2018)
23. Li, X.D., Branke, J., Kirley, M.: On performance metrics and particle swarm methods for dynamic multiobjective optimization problems. In: *proceedings of 2007 IEEE Congress on Evolutionary Computation*, Singapore, pp. 576–583, IEEE (2007)