A Shift Vector Guided Multiobjective Evolutionary Algorithm Based on Decomposition for Dynamic Optimization

ZHENG ZHU*, XIANG TIAN, CHANGGAO XIA, LONG CHEN*, AND YINGFENG CAI*

Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China

Corresponding author: Yingfeng Cai (caicaixiao0304@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51805222 and Grant U1764257, in part by the National Key Research and Development Project under Grant 2016YFD0700402, in part by the China Postdoctoral Science Foundation under Grant 2018M632247, in part by the Key Project for the Development of Strategic Emerging Industries of Jiangsu Province under Grant 2016-1094, and in part by the National Natural Foundation of Jiangsu Province under Grant BK20180100.

ABSTRACT This paper presents a novel algorithm to deal with dynamic multiobjective optimization problems, in which the objective functions change over time. The algorithm adopts the decomposition framework to decompose the multiobjective optimization problems into a number of scalar optimization subproblems. For each subproblem, its respective solutions obtained in several former consecutive environments can form a moving trajectory over time. A shift vector guided prediction model is proposed, which samples three immediately previous solutions of each subproblem to construct two shift vectors. The shift vectors use the weighted summation to generate a new shift vector as the forthcoming motion of the target solution. Then a new location in the later environment is estimated based on the current location and the newly generated shift vector. When detecting an environmental change, the multiobjective evolutionary algorithm based on decomposition will update the population using the predicted solutions by the proposed model. Empirical results demonstrate that our proposed algorithm is effective in tracking dynamic optimal solutions and shows great superiority comparing with state-of-the-art methods.

INDEX TERMS Decomposition, dynamic environments, multiobjective evolutionary algorithm, prediction, shift vector.

I. INTRODUCTION Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) [1] is a typical approach to deal with Multiobjective Optimization Problems (MOPs), which decomposes a MOP into a number of scalar optimization subproblems and optimizes them simultaneously. In MOEA/D, each subproblem is optimized by only using information from its several neighboring subproblems [1]. MOEA/D has attracted significant attention from researchers in the research area of Evolutionary Computation (EC) [2] since it was proposed. The performance of MOEA/D on MOPs has been improved by proposing novel weight vector generation methods [3], decomposition approaches [4], reproduction operator [5], [6], [7], mating selection [8] and replacement mechanism [9], etc. Furthermore, the decomposition-based framework has been extended to the constrained multiobjective optimization [10]–[12], many-objective optimization [13], [14], dynamic multiobjective optimization [15], etc. Among the published decomposition-based algorithms, many of them are proposed for many-objective optimization, only a few of them attempt to deal with dynamic multiobjective optimization problems [16]–[18].

One of the essential characteristics of Dynamic Multiobjective Optimization Problems (DMOPs) is that the objective functions, constraints, or decision variables change over time, which results in varying the Pareto-Optimal Set (POS) and/or Pareto-Optimal Front (POF) [19]. Many real-world MOPs are dynamic in nature, e.g., dynamic multiobjective job shop scheduling [20], online optimization of shift strategy for hydro-mechanical continuously variable transmission [21], control of time-varying unstable plants [22]. Due to the presence of dynamics, the optimization of DMOPs is much more challenging. The algorithm must not only approximate the POS and POF in a short time period, but also track the varying POS and/or POF [23]. The state-of-the-art methods are mainly developed from Nondominated Sorting Genetic
Algorithm II (NSGA-II) [24], while only a few of them are extended from MOEA/D [15], [25], [26]. Randomly reinitializing the population, retrieving the historical information, and predicting the new solutions are the typical strategies to be incorporated into the original Multiobjective Evolutionary Algorithms (MOEAs). Although there are several algorithms that using MOEA/D to solve DMOPs by the above strategies, the particular characteristic of MOEA/D is not fully exploited. More simple and easy-understood strategies should be developed to improve the performance of MOEA/D on DMOPs.

This paper presents a novel multiobjective evolutionary algorithm based on decomposition for DMOPs. A DMOP can be divided into a series of MOPs according to the interval of change, in which a MOP can be considered static. A MOP is decomposed into several scalar optimization subproblems, which are formulated by decomposition approach—Tchebycheff using uniformly distributed weight vectors. The solution of each subproblem is represented by the individual of population, which is allocated by the weight vectors. For each subproblem, its solutions obtained in several consecutive environments can form a moving trajectory over time. A shift vector guided prediction model is proposed, which samples three immediately previous solutions of each subproblem to construct two shift vectors. We assume that the later shift vector are linearly corrected to the former two vectors. Thus the shift vectors use the weighted summation to generate a new shift vector as the forthcoming motion of that solution. Then a new location (solution) in the later environment is estimated based on the current location and the newly generated shift vector. This prediction model provides an initial population for the algorithm, after detecting an environmental change. In order to reuse the knowledge acquired from the most previous environment, half of population members are retained to the new population, and the others are predicted their new locations using the shift vector guided prediction model. The experiments show that the proposed shift vector guided multiobjective evolutionary algorithm based on decomposition (MOEA/D-SV) is effective in tracking dynamic optima, and outperforms some state-of-the-art methods on the benchmark tested.

The remainder of this paper is organized as follows. Section 2 presents the background of this paper, including the definition of DMOP and related work. In Section 3, we present the proposed algorithm. Experimental design is given in Section 4. A comparative study and empirical results are presented in Section 5. Finally, the conclusion is given in Section 6.

II. BACKGROUND

A. PROBLEM DEFINITION

DMOPs can change over time in various ways. In this paper, we only focus on the following type of DMOPs:

$$\begin{align*}
\min F(x, t) &= (f_1(x, t), f_2(x, t), \cdots, f_m(x, t))^T \\
\text{subject to } x &\in \Omega
\end{align*}$$

where $m$ is the number of objectives, $t = 0, 1, 2\cdots$ represents discrete time instants, $x$ is the decision variable vector, and $\Omega$ represents the decision space. The objective vector $F(x, t)$ consists of $m$ time-varying objective functions that change intermittently.

**Definition 1 (Pareto Dominance):** At time $t$, a solution $x_1 \in \Omega$ Pareto dominates a solution $x_2 \in \Omega$, denoted by $x_1 \succ x_2$, if and only if [16]:

$$\begin{align*}
\forall j \in \{1, \cdots, m\}, f_j(x_1, t) &\leq f_j(x_2, t) \\
\exists j \in \{1, \cdots, m\}, f_j(x_1, t) &< f_j(x_2, t)
\end{align*}$$

**Definition 2 (Pareto-Optimal Set):** If a solution $x^* \in \Omega$ is said to be nondominated at time $t$ if and only if there is no other solution $x \in \Omega$ such that $x \succ x^*$ at time $t$. The Pareto-Optimal Set (POS) is the set of all Pareto-optimal solutions at time $t$, that is:

$$\text{POS} = \{x^* \in \Omega | \nexists \, x \in \Omega, x \succ x^*\}$$

**Definition 3 (Pareto-Optimal Front):** At time $t$, the Pareto-Optimal Front (POF) is the corresponding objective vectors of the POS:

$$\text{POF} = \\{F(x^*, t) | x^* \in \text{POS}\}$$

DMOPs require a multiobjective evolutionary algorithm to obtain a set of solutions as closely as possible to the POF before it may change. Meanwhile, the set of solutions should be as diverse as possible.

DMOPs can be classified into four types as follows:

- **Type I:** The POS changes whereas the POF remains static.
- **Type II:** Both POS and POF change.
- **Type III:** The POS does not change whereas the POF does.
- **Type IV:** Both POS and POF remain static, although the problem changes.

In this paper, we focus on the first three types of changes.

B. RELATED WORK

Applying MOEAs to solve DMOPs has attracted much attention from the EC community, and a variety of algorithms have been proposed that were mostly extended from NSGA-II [24], RE-MOEA [27], or MOEA/D [1]. When employing MOEAs to DMOPs, two issues should be addressed—diversity loss and convergence acceleration. For the population-based optimization algorithms, convergence during iterations may cause a lack of diversity. Thus, the algorithm loses its ability to flexibly react to changes. For this reason, the strategy of increasing diversity after a change detection has been proposed. Due to the time limit, a regular population may not obtain a near-optimal POS before the change. Thus, many researchers proposed to use knowledge of the previous search in order to accelerate optimization after a change. Memory-based approaches and prediction-based approaches are commonly used to accelerate the convergence after a change detection.
1) INCREASING OR MAINTAINING DIVERSITY

Deb et al. [28] extended NSGA-II to solve a real-world dynamic optimization of a hydro-thermal power scheduling problem. In the algorithm, a few random solutions or a few mutated solutions were introduced into the population as soon as there was a change in the problem. Zeng et al. [29] introduced an orthogonal MOEA for DMOPs. The algorithm used the previous solutions obtained before the change as the initial population after the change. It applied two crossover operators, namely, orthogonal crossover that was based on an orthogonal design and linear crossover. One of them was randomly selected and applied to two randomly selected parents to produce offspring. Zheng [30] proposed a MOEA adapted for DMOPs, in which hypermutation was used to preserve a certain number of elitist individuals, while the rest of the individuals were replaced by randomly created new individuals.

2) MEMORY-BASED APPROACHES

Wang and Li [31] proposed three memory-based schemes that were incorporated into NSGA-II to handle DMOPs. The explicit memory scheme was used to store a few solutions from the past POS, a local-search operator was used to provide a neighborhood solution of the archive solution to the new population, and a hybrid memory scheme mixed explicit and local-search memory together. Goh and Tan [32] presented a co-evolutionary multiobjective algorithm based on competitive and cooperative mechanisms to solve DMOPs. In order to adapt to dynamic environments, the external population denoted as the temporal memory was used in addition to the archive in order to store the potentially useful information about past POF since that the archive solutions were discarded at the new environment. Chen et al. [15] integrated a memory-enhanced multiobjective evolutionary algorithm based on decomposition with a stable matching model to solve DMOPs. In the algorithm, a subproblem-based bunchy memory scheme was used to store good solutions from the previous environments to retrieve them after detecting a change. Sahmoud and Topcuoglu [33] presented a new hybrid strategy by integrating the memory scheme with NSGA-II. The algorithm utilized an explicit memory to store a number of non-dominated solutions in order to retrieve them to reinitialize part of the population after detecting an environmental change. Azzouz et al. [34] proposed an adaptive hybrid population management strategy using memory, local search, and random strategies to handle DMOPs, in which the memory strategy made use of past optimal solutions.

3) PREDICTION-BASED APPROACHES

Hatzakis and Wallace [35] proposed a forward-looking approach using evolutionary algorithms for DMOPs. The algorithm combined a forecasting method with a MOEA, in which an autoregressive model was used as the forecasting method that was based on the sequence of prior optimum locations. Zhou et al. [36] proposed a population prediction strategy for DMOPs, in which a Pareto-optimal set was divided into a center point and a manifold. They also used the autoregressive model to predict the new center by a sequence of center points. Wu et al. [37] proposed a directed search strategy to improve the performance of NSGA-II in dynamic environments. The strategy reinitialized the population based on the predicted moving direction as well as the directions that were orthogonal to the moving direction of POS after detecting a change. Muruganantham et al. [25] used the Kalman Filter model to predict the solutions, which was incorporated with MOEA/D to solve DMOPs. Jiang and Yang [38] presented a steady-state and generational evolutionary algorithm that combined the fast and steadily tracking ability of steady-state algorithms and good diversity preservation of generational algorithms for handling DMOPs. In their algorithm, a prediction approach was used to forecast the moving direction and the movement step-size of solutions. Zou et al. [39] proposed a prediction strategy based on center points and knee points. A forward-looking center points are used to predict the non-dominated solutions, and the knee point set was introduced to the predicted population to predict accurately the location and distribution of the Pareto front after detecting a change. Ruan et al. [40] proposed a hybrid diversity maintenance method to improve prediction accuracy. The moving direction of the center points were predicted to relocate a number of solutions close to the new Pareto front. The gradual search was used to produce some well-distributed solutions in the decision space so as to compensate for the inaccuracy of the predicted solutions. Jiang et al. [41] integrated transfer learning and population-based evolutionary algorithms to solve DMOPs. In their algorithm, a domain adaptation method called transfer component analysis was adopted to construct a prediction model to use the gained knowledge of finding POS to generate an initial population for the optimization function at the later time period.

III. PROPOSED ALGORITHM

A. DECOMPOSITION

MOEA/D decomposes the problem (1) with a fixed time instant into a number of single-objective optimization subproblems through an aggregation function. A Pareto-optimal solution to a MOP, under mild conditions, is an optimal solution of a single-objective optimization problem whose objective is a weighted aggregation of all the individual objectives. In this paper, the Tchebycheff approach acts as the decomposition method. Let \( \lambda_1, \lambda_2, \ldots, \lambda_N \) be a set of even spread weight vectors, a multiobjective optimization problem at time \( t \) can be decomposed into \( N \) scalar optimization problems, and the \( i \)-th subproblem \( (i = 1, \ldots, N) \) at time \( t \) is given by:

\[
\begin{align*}
\text{minimize} & \quad g^i \left( x \left| \lambda^i, z^* \right. \right) = \max_{1 \leq j \leq m} \{ \lambda_j^i \left| f_j (x, t) - z_j^* \right| \} \\
\text{subject to} & \quad x \in \Omega
\end{align*}
\]

where \( z^* = (z_1^*, \ldots, z_m^*)^T \) is the reference point, i.e., \( z_j^* = \min \{ f_j (x, t), x \in \Omega \} \) (for a minimization problem) for
each \( j = 1, \cdots, m \). MOEA/D minimizes these \( N \) subproblems simultaneously in a single run.

In MOEA/D, a neighborhood of weight vector \( \lambda^i \) is defined as a set of its several closest weight vectors in \( \{ \lambda^1, \cdots, \lambda^N \} \). The neighborhood of the \( i \)-th subproblem consists of all the subproblems with the weight vectors from the neighborhood of \( \lambda^i \). A population of \( N \) solutions is randomly generated and each solution is randomly allocated to a particular subproblem. Note that, the objective functions change over time in DMOPs, but the weight vectors do not change.

**B. SHIFT VECTOR GUIDED PREDICTION**

For the \( i \)-th subproblem (\( i = 1, \cdots, N \)) at time \( t \), its solution can be obtained by any single-objective optimization algorithm. The different solution of the \( i \)-th subproblem at different time windows (environments) can be found as long as enough evaluations were given before changing the objective functions. However, the problems may change quickly in the real-world optimization problems, no enough evaluations may be allowed. Therefore, developing an approach to accelerate the convergence speed is extremely important for solving DMOPs. On the one hand the convergence of a MOEA should be improved in a static environment, but on the other the knowledge obtained in the previous environments should be fully exploited to provide a good initial population for a MOEA in the new environment. Generally, the objective functions of problems change over time with some regularities [38]. In addition, in many cases, the problems change smoothly. Therefore, the obtained solutions in the previous environments can be used to predict new solutions in the new environment.

For each subproblem, its corresponding solutions obtained by the optimization algorithm in several consecutive environments can form a moving trajectory over time. Fig.1 illustrates the decomposition of a dynamic multiobjective optimization problem in the objective space. For the subproblems with the weight vectors \( \lambda^1, \lambda^2, \lambda^3 \), the optimization algorithm approximates the corresponding Pareto-optimal solutions at the time windows T, T-1, T-2, respectively. A series of these historical solutions can be used to predict their respective new solution in the next time window, which can be considered as a time series forecasting problem. Theoretically, any time series model can be used to make a prediction for the new solution of each subproblem. However, there occurs errors of the historical data (solutions obtained) since the algorithm may not converge properly within the allowed evaluations. The approximated solutions may deviate from the path under the unknown governing dynamics, which causes to the errors of the input data for the prediction model. Accordingly, an accurate prediction model may not forecast well for the new solution of each subproblem. In this paper, we design a raw prediction model to estimate a new solution for each subproblem.

As shown in Fig.1, there are three historical solutions for each subproblem at the end of the time window \( T \). We try to use these data to make a prediction for the Pareto-optimal solution of each subproblem in the next time window. For the \( i \)-th subproblem (\( i = 1, \cdots, N \)) at the time window \( T \), its solution obtained by the algorithm is denoted as \( x^i_T \), and the position of this solution in the decision space is also denoted as \( x^i_T \). Positions in two consecutive time windows can form a shift vector: \( x^i_T - x^i_{T-1} \), which indicates the motion of this solution between these time windows. We use the previous shift vectors to predict the forthcoming motion of the solution between the time window \( T \) and \( T+1 \). The prediction can be expressed as follows:

\[
\Delta x^i_T = \alpha^x * x^i_T - x^i_{T-1} + (1 - \alpha^x) * x^i_{T-1} - x^i_{T-2} \quad (6)
\]

\[
x^i_{T+1} = x^i_T + \Delta x^i_T \quad (7)
\]

where \( \Delta x^i_T \) is the predicted shift vector of the solution \( x^i_T \) at the end of the time window \( T \), which indicates that the solution will move as this shift vector to the new location at the time window \( T+1 \). \( \alpha^x \) is a uniformly random number generated between 0 and 1. \( x^i_T, x^i_{T-1}, \) and \( x^i_{T-2} \) are solutions obtained at the time window \( T, T-1, \) and \( T-2, \) respectively. \( x^i_{T+1} \) is the predicted solution that is expected to approximate the true Pareto-optimal solution of the \( i \)-th subproblem at the time window \( T+1 \).

The proposed prediction approach only needs three historical solutions to construct the shift vectors. The forthcoming motion is assumed to be affected by the previous shifts. Fig.2 illustrates the prediction of \( x^i_T \)'s new location in a
2-D decision space at the time window $T+1$. $\overrightarrow{v_i}$ represents the shift vector: $x_i^T - x_i^{T-1}$, and $\overrightarrow{v}$ represents $x_i^{T-1} - x_i^{T-2}$. $\Delta x_i^T$ is the predicted shift vector of $x_i^T$ at the end of the time window $T$, which is computed by the equation (6). $x_i^{T+1}$ is the predicted solution of the $i$-th subproblem at the time window $T+1$. Guided by the previous shift vectors, the forthcoming motion of each solution $x_i^T$ ($i = 1, \cdots, N$) can be predicted and meanwhile their respective new location in the decision space at the next time window can be estimated.

Algorithm 1 Reaction to Change

1. **Input:** Pop$^T$, Pop$^{T-1}$, Pop$^{T-2}$, $N$;
2. **for** $i = 1$ to $N$ do
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. **end for**
12. **Output:** Pop$^{T+1}$.

C. REACTION TO CHANGE
After predicting the new locations of these solutions in the next time window, the population should be updated promptly. As mentioned before, the prediction model that we designed is a raw approach to make prediction, due to the existence of errors between the solutions obtained and the true Pareto-optimal solutions. Accordingly, the predicted solutions may not be very close to the true Pareto-optimal solutions. Generally, the problems change smoothly in the fitness landscape, the solutions in the decision space at two consecutive environments may be not far away in the Euclidean distance. Hence, the most recent solutions may be useful for tracking the new locations in the decision space at the later environment. Therefore, 50% of the current population members are retained, and the others are predicted their new locations. The pseudocode of updating the population is given in Algorithm 1, in which Pop$^T$ represents the population that contains the solutions obtained at the time window T, $N$ is the population size. The predicted solutions and the retained solutions are assigned as every other member, in order to uniformly distribute them to the neighborhood of each subproblem. Note that, the prediction scheme works beginning from the fourth time window. The prediction model is not available before that, due to limited historical information, therefore all solutions are retained to the population.

D. FRAMEWORK OF MOEA/D-SV
The whole framework of MOEA/D-SV is given in Algorithm 2. Note that, the proposed algorithm is based on the framework of MOEA/D [5] that published in 2009, in which a differential evolution (DE) operator and a polynomial mutation operator are used for producing new solutions. Comparing with the original framework of MOEA/D, Step 2 is the additional part: detection and response for the environmental change. 10% of population members are chosen for re-evaluation to detect environmental changes.
which is commonly used by many algorithms [36], [25], [38]. In addition, there are two more modifications in this framework. In the reproduction operator, we randomly select three individuals from the mating pool to reproduce a new solution. While two individuals are randomly selected from the mating pool, in the original framework of MOEA/D. In Step 3.6, each member in the mating pool is compared with the newly generated solution, and be replaced if it is worse than the new solution in the weighted aggregation function. But in the original framework of MOEA/D, only \( n_r \) members in the mating pool are compared with the new solution (\( n_r \) is set to be two in [5]).

### IV. EXPERIMENTAL DESIGN

#### A. BENCHMARK AND PERFORMANCE METRIC

In order to evaluate the performance of MOEA/D-SV, 16 benchmark problems are tested, including four FDA test suite [42] (FDA1-FDA4), four JJZ test suite [36] (F5-F8), and eight JY test suite [43] (JY1-JY8). Note that, FDA4 and F8 are triobjective optimization problems, and the others are biobjective optimization problems. The time instance \( t \) involved in these problems is defined as \( t = 1/n_r \times \lfloor \tau / \tau_r \rfloor \) (where \( n_r \), \( \tau \), and \( \tau_r \) represent the severity of change, the frequency of change, and the iteration counter, respectively). The dimension of decision variables in all these problems is set as 10. Note that, the dimension of FDA2 is set to be \( |X_\| = 4 \), \( |X_{\|} = 5 \), and the dimension of FDA3 is set to be \( |X_\| = 2 \), \( |X_{\|} = 8 \). The definitions of these problems can refer to the respective literature [28], [36], [42], [43].

We use two metrics to evaluate algorithms: Inverted Generation Distance (IGD) and its modified version MIGD.

Let \( Q^* \) be a set of uniformly distributed points in the true POP\(^t\), and \( Q^t \) be an approximation of POP\(^t\). The IGD metric is defined as:

\[
IGD(Q^*, Q^t) = \sum_{v \in Q^t} d(v, Q^*) / |Q^*|
\]

where \( d(v, Q^*) = \min_{u \in Q} \| F(v) - F(u) \| \) is the distance between \( v \) and \( Q^t \), and \( |Q^*| \) is the cardinality of \( Q^* \). The IGD metric can measure both diversity and convergence. In the experiments, 500 points and 1000 points are uniformly sampled from the true POP\(^t\) of biobjective problems and triobjective problems, respectively, for computing the IGD metrics.

The MIGD metric is defined as the average of the IGD values in some time windows over a single run:

\[
MIGD = \frac{1}{|T|} \sum_{t \in T} IGD(Q^*, Q^t)
\]

where \( T \) is a set of discrete time instances in a run and \(|T|\) is the cardinality of \( T \).

The lower the MIGD value is, the better the overall performance of the algorithm. In the following empirical results, the best mean values are highlighted in bold. In order to compare statistically, Wilcoxon’s rank sum test at the 5% significance level is conducted to indicate the significance of difference between two algorithms.

#### B. COMPARED ALGORITHMS AND PARAMETER SETTINGS

In order to show the superiority of MOEA/D-SV, four state-of-the-art algorithms are compared in the following empirical studies, including MOEA/D [5], MOEA/D-KF [25], DNSGA-II-A [28], and DSS [37]. The original MOEA/D is modified to address DMOPs, which is denoted as MOEA/D-RI. A fraction of its population is randomly reinitialized after detecting the environment changes. These four algorithms use the DE operator and the polynomial mutation operator to generate new individuals. The problems and algorithm parameter settings are as follows.

1) **BENCHMARK SETTINGS**

To study the impact of change frequency and change severity, a variety of parameters are used. The severity of change is \( n_r = 10, 5 \). The frequency of change is \( \tau_r = 5, 10 \).

2) **COMMON PARAMETERS IN ALL ALGORITHMS**

The population size is set as 100 and 300 for biobjective problems and triobjective problems, respectively. \( CR = 0.5, F = 0.5 \) in the DE operator. \( \eta = 20, p_m = 1/n \) in the polynomial mutation operator (\( n \) is the number of variables).

3) **PARAMETERS IN MOEA/D**

To evaluate the MOEA/D-based algorithms fairly, some parameters in MOEA/D-SV, MOEA/D and MOEA/D-KF are set the same: the neighborhood size is set as 20, \( \delta = 0.8 \), \( n_r = |P| \). P is the mating pool, from which individuals are selected to produce a new solution. For details about P, please see [5].

4) **OTHER PARAMETERS**

In MOEA/D and DNSGA-II-A, 20% of population members are randomly reinitialized within the domain when the environment changes. The number of total generations in each run is fixed to be \( 40 \times \tau_r \). Each algorithm is run independently 20 times for each test instance.

### V. EXPERIMENTAL RESULTS

#### A. RESULTS ON FDA

Comparison results of MOEA/D-SV with the other algorithms on FDA1-FDA4 in terms with MIGD are shown in Table 1. Observing from Table 1, MOEA/D-SV performs best among these algorithms on all test cases, in terms of the mean values of MIGD. A smaller value of \( n_r \) results in the problems changing in a larger severity, which causes the algorithms to track the changing POS and/or POF much harder. The MIGD values obtained on the problems with \( n_r = 5 \) are worse. A smaller value of \( \tau_r \) means that the problems changed more frequently, the algorithms are given less evaluations to approximate the Pareto-optimal solutions before the environmental changes. Therefore, the MIGD
values obtained on the problems with $\tau_t = 10$ are better than the MIGD obtained on the problems with $\tau_t = 5$. The Pareto-optimal solutions of FDA change over time but the POF remains stationary. In FDA2, the POF changes from convex to non-convex shapes in the objective space over time and a subset of the decision variables also changes. For these two problems, MOEA/D-SV and MOEA/D-RI obtain quite well results, which have evident superiority comparing with the other three algorithms. FDA3 is a problem in which the changes shift the POS and affect the density of points on the POF, which definitely challenges these algorithms to track the varying POS and POF. The MIGD values obtained on FDA3 are obviously worse comparing with the results on FDA1 and FDA2. FDA4 is a dynamic triobjective optimization problem, in which only the POS changes over time. The three MOEA/D-based algorithms perform similar, and the results obtained by MOEA/D-SV are only a little better than that of MOEA/D-RI and MOEA/D-KF. In addition, these three algorithms seem less sensitive to the severity of change on FDA4.

Fig. 3(a)-(d) present the IGD values computed over generations that obtained by five algorithms on the FDA test suites with $n_t = 5$, $\tau_t = 10$, showing the average values over 20 runs. In order to show the tracking process of these algorithms clearly, only the IGD values over the first 100 generations are presented, in which 10 environmental
TABLE 2. Mean and standard deviations of migd values obtained by five algorithms on ZJZ.

| Problems | \(n_t, \tau_t\) | MOEA/D-SV | MOEA/D-RI | MOEA/D-KF | DNSGA-II-A | DSS |
|----------|----------------|-----------|-----------|-----------|-------------|-----|
| F5       | (10,5)         | 0.5264±0.1065 | 0.7442±0.0685 (↓) | 0.6636±0.2141 (↓) | 2.0312±0.0725 (↑) | 0.6665±0.0272 (↑) |
|          | (10,10)        | 0.2317±0.0490  | 0.6139±0.0585 (↓) | 0.2735±0.0680 (↓) | 0.8132±0.1080 (↑) | 0.3602±0.0200 (↑) |
|          | (5,5)          | 1.1811±0.0892  | 1.3578±0.0685 (↓) | 1.1965±0.7040 (↓) | 3.0554±0.0942 (↑) | 1.5445±0.2482 (↑) |
|          | (5,10)         | 0.8044±0.0721  | 1.1074±0.0845 (↓) | 0.8132±0.0585 (↓) | 1.4362±0.0752 (↑) | 0.8525±0.0755 (↑) |
| F6       | (10,5)         | 0.3886±0.0542  | 0.4892±0.0256 (↓) | 0.4925±0.1312 (↑) | 1.4025±0.0224 (↑) | 1.0288±0.0140 (↑) |
|          | (10,10)        | 0.2586±0.0378  | 0.3281±0.0205 (↓) | 0.2825±0.0755 (↓) | 0.7285±0.0582 (↑) | 0.3380±0.0072 (↑) |
|          | (5,5)          | 0.9638±0.0240  | 0.9822±0.0725 (↓) | 0.9948±0.0345 (↓) | 2.1286±0.0301 (↑) | 1.8152±0.0361 (↑) |
|          | (5,10)         | 0.7151±0.0159  | 0.7193±0.0384 (↓) | 0.6237±0.0685 (↑) | 1.2182±0.0440 (↑) | 0.7841±0.0712 (↑) |
| F7       | (10,5)         | 0.2869±0.0135  | 0.4732±0.0524 (↓) | 0.5923±0.2632 (↓) | 1.5054±0.0375 (↑) | 1.0882±0.0838 (↑) |
|          | (10,10)        | 0.2229±0.0252  | 0.3585±0.0350 (↓) | 0.2981±0.0204 (↓) | 0.7436±0.0602 (↑) | 0.3612±0.0330 (↑) |
|          | (5,5)          | 0.4814±0.0142  | 0.6332±0.0553 (↓) | 0.5882±0.0956 (↓) | 2.0825±0.0520 (↑) | 1.9850±0.0815 (↑) |
|          | (5,10)         | 0.4321±0.0257  | 0.5116±0.0748 (↓) | 0.5793±0.0240 (↓) | 1.1395±0.0664 (↑) | 0.8205±0.1064 (↑) |
| F8       | (10,5)         | 0.1067±0.0021  | 0.1204±0.0030 (↓) | 0.1415±0.0007 (↓) | 0.2636±0.0028 (↑) | 0.2824±0.0007 (↑) |
|          | (10,10)        | 0.0801±0.0050  | 0.0952±0.0045 (↓) | 0.1073±0.0042 (↑) | 0.1554±0.0035 (↑) | 0.1802±0.0042 (↑) |
|          | (5,5)          | 0.1192±0.0010  | 0.1416±0.0008 (↓) | 0.1470±0.0004 (↓) | 0.4112±0.0014 (↑) | 0.3502±0.0192 (↑) |
|          | (5,10)         | 0.0866±0.0018  | 0.1092±0.0032 (↓) | 0.1087±0.0038 (↓) | 0.2223±0.0022 (↑) | 0.2354±0.0005 (↑) |

“−”, “−” and “−” indicate MOEA/D-SV performs significantly better or worse than or not significantly different from the corresponding algorithm, respectively.

changes happen. Observing from Fig.3(a), the three MOEA/D-based algorithms have much better final IGD values than the two NSGA-II-based algorithms in the first three environments (time windows). Starting from the fourth environment, the shift vector guided prediction scheme works. MOEA/D-SV has a lowest IGD value in the beginning of the fourth environment (the 31-th generation), which illustrates that the initial population of MOEA/D-SV at this time window is much closer to the true POF in the objective space. Since of good initial population provided by the shift vector guided prediction model, MOEA/D-SV can better approximate the true POF in the objective space and obtain a much lower IGD value in the end of the four environment. The other algorithms seem no clear effect of providing good initial population. In the later time windows, MOEA/D-SV has the best IGD values in the initial stage and the end stage for the most of the time windows. In Fig.3(b) and (d), MOEA/D-SV has the best IGD values in the initial stage and the end stage for the most of the time windows. In Fig.3(b) and (d), MOEA/D-SV has the best IGD values in the initial stage and the end stage for the most of the time windows. But in Fig.3(c), MOEA/D-SV does not obtain the lowest IGD values in the middle time windows. It performs unstably for the response to the changes. This situation also happens on the other algorithms, which illustrates that FDA3 greatly challenges these algorithms to respond to the change. Viewing from Fig.3, MOEA/D-SV is very capable of tracking the environmental changes generally.

B. RESULTS ON ZJZ

Unlike the FDA, the ZJZ test suites (F5-F8) have nonlinear correlation between decision variables. In addition, the POS and POF change over time with strongly-nonlinear dynamic characteristics in the decision space and objective space, respectively. Therefore, the ZJZ problems are much difficult to solve for MOEAs especially when the problem changes frequently and severely. Table 2 presents the results obtained by five algorithms on F5-F8. The average MIGD values obtained by these algorithms on the ZJZ problems are clearly worse than those obtained on the FDA, in general. Among these algorithms, MOEA/D-SV performs best and has evident superiority on most cases of F5-F8 with different change severities and frequencies. For F5 and F6, MOEA/D-KF performs a little worse than MOEA/D-SV on most cases, but beats MOEA/D-SV on F6 with \( n_t = 5, \tau_t = 10 \). For F7 and F8, MOEA/D-SV outperforms the other algorithms obviously.

The tracking characteristics of these algorithms on ZJZ are presented in Fig.4, in which the average IGD values over the first 100 generations on ZJZ test suites with \( n_t = 5, \tau_t = 10 \) are plotted. Observing from Fig.4(a)-(c), the final IGD values of these algorithms on the end stage of each time window fluctuate a lot, due to the influence of the strongly-nonlinear dynamic characteristics of these problems. But the IGD values obtained by MOEA/D-SV in the initial stage of most of the time windows are still better than most of the other algorithms, which illustrates that the proposed prediction model is useful and better than the other approaches. The domain of decision variables of F8 is smaller than that of F5-F7, and the nonlinear characteristics of dynamic changes of F8 are weaker than that of F5-F7. Therefore, the tracking curves of these algorithms on F8 are relatively smooth comparing with the curves in Fig.4(a)-(c). The IGD values obtained by MOEA/D-SV in the initial stage of each environment are almost the lowest among these algorithms, which benefits from the good initial population provided by the proposed prediction model in this paper.

C. RESULTS ON JY

Table 3 presents the results acquired by five algorithms on the JY test suites. Solving JY1 and JY2 is relatively simple, because there is no correlation between decision variables and the Pareto-optimal solutions change over time with a weakly-nonlinear pattern. These two problems can assess
the convergence speed and the reactivity of an algorithm. MOEA/D-SV outperforms MOEA/D-RI and MOEA/D-KF on these two problems with different change severities and frequencies. In JY3, POS and POF change over time with a
FIGURE 5. The average IGD values over generations on JY test suites with $n_t = 5$, $\tau_t = 10$.

strongly-nonlinear pattern. MOEA/D-SV has the similar performance to MOEA/D-RI, which illustrates that the proposed prediction model did not provide a good initial population to approximate the new POF in the new environment. The proposed prediction model may not well estimate for this kind of problem like JY3. JY4 has a time-varying number of disconnected POF segments, (i.e. the POF is discontinuous). However, there is no correlation between decision variables in JY4. A shown in Table 3, the results are less sensitive to the change severity, especially for the three MOEA/D-based algorithms. MOEA/D-SV still has weak superiority on this problem. The POF of JY5 changes from convex geometry to concave geometry, whereas the POS of JY5 remains stationary. MOEA/D-SV and MOEA/D-RI perform very well on this problem. Even MOEA/D-RI performs better than MOEA/D-SV, because 80% of population in MOEA/D-RI retained from the old solutions obtained in the previous environment, whereas only 50% of population in MOEA/D-SV...
retained the old solutions. JY6 are JY7 multimodal problems, where both the POF and the POS dynamically change over time. Since of existing local optima, more evaluations are needed to approximate the Pareto-optimal solutions before the environment changes. Therefore, the results on JY6 and JY7 are much worse comparing with the results on the above JY test suites. MOEA/D-SV still has superiority on these two problems. The POS of JY8 remains static, whereas the POF varies over time. These algorithms (except DSS) perform similar, because the old solutions were retained to the new population after the change in these algorithms. MOEA/D-SV has weak superiority when the problem changes frequently (τf = 5), whereas DNSGA-II-A has better performance when τf = 10.

The tracking characteristics of these algorithms on JY test instances are presented in Fig.5, in which the average IGD values over the first 100 generations on JY test suites with nL = 5, τf = 10 are plotted. MOEA/D-SV and MOEA/D-KF have better IGD values in the initial stage of each environment on JY1 and JY2, which illustrates that the proposed shift vector guided model and the Kalman filter model are useful for providing good initial population to adapt to the new environment. MOEA/D-SV and MOEA/D-RI track the changes stably and have similar tracking abilities on JY3. Starting from the fourth time window of JY4, MOEA/D-SV has the relatively stable IGD values, due to the good tracking ability. Since the POS of JY5 remains static, MOEA/D-SV and MOEA/D-RI perform best. There is no one can track stably on JY6 and JY7, because of the complex variation of their POF. But MOEA/D-SV still has the lowest IGD values in the initial stage of most of the time windows, which illustrates the effect of the shift vector guided prediction model.

VI. DISCUSSION AND CONCLUSION
This paper presents a novel algorithm to solve dynamic multiobjective optimization problems, which is extended from the typical static multiobjective evolutionary algorithm——MOEA/D. We proposed a shift vector guided model to predict a new location of the solution of the subproblem in the later new environment. The model samples three intermediately previous locations of that solution to construct two shift vectors, which used the weighted summation to generate a new shift vector as the forthcoming motion of that solution. Then a new location in the later environment is estimated based on the current location and the estimated shift vector. The shift vector guided model works when detecting the environmental change, but not the all solutions are predicted their new locations. Half of them are still remain their old locations to explore in the new environment. Empirical results demonstrate that our proposed MOEA/D-SV has clear superiority over some state-of-the-art algorithms on most of benchmark problems tested.

The main novelty and contribution of this algorithm is that it proposed a simple and easily-understood prediction model to incorporate into the multiobjective evolutionary algorithm based on decomposition to adapt to the dynamic environments. The proposed model did not make any hypothesis to limit the scope of application for DMOPs, as long as the objective functions in the problem change over time as regular rules. In addition, there is no additional parameters needed to tune, on the basis of MOEA/D. Note that, half of the current population members were remained to the new population after detecting the environmental change. The proportion of the old solutions in the new population will not excessively affect the performance of tracking the varying POS and POF. We do not show the experimental results about this part, due to the space limit. Remaining too many old solutions is only useful for solving the problems in which the POS remains stationary in the dynamic environments. However, we have no prior knowledge of the dynamic feature of the problems. The shift vector prediction model just provides the raw data for the population. It may deviate the true data. A few old solutions are needed and may be better than some predicted solutions. Therefore, half of the population members remain the old solutions and the others accept the predicted locations whenupdating the population after detecting the environmental change.

Our proposed MOEA/D-SV works for those DMOPs whose objective functions change over time smoothly and with some regularities. However, its performance may be affected if the objective functions change over time with some complex regularities, e.g., the decision variables of problems change with strongly-nonlinear patterns or change severely. This should be improved in our future work.

REFERENCES
[1] 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.
[2] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, “A survey of multiobjective evolutionary algorithms based on decomposition,” IEEE Trans. Evol. Comput., vol. 21, no. 3, pp. 440–462, Sep. 2017.
[3] H. Li, M. Ding, J. Deng, and Q. Zhang, “On the use of random weights in MOEA/D,” in Proc. IEEE Congr. Evol. Comput. (CEC), May 2015, pp. 978–985.
[4] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, “Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm,” IEEE Trans. Evol. Comput., vol. 20, no. 3, pp. 475–480, Jun. 2016.
[5] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II,” IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 284–302, Apr. 2009.
[6] K. Li, A. Fialho, S. Kwong, and Q. Zhang, “Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition,” IEEE Trans. Evol. Comput., vol. 18, no. 1, pp. 114–130, Feb. 2014.
[7] L. Ke, Q. Zhang, and R. Battiti, “MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony,” IEEE Trans. Cybern., vol. 43, no. 6, pp. 1845–1859, Dec. 2013.
[8] S. Jiang and S. Yang, “An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts,” IEEE Trans. Cybern., vol. 46, no. 2, pp. 421–437, Feb. 2016.
[9] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, “Stable matching-based selection in evolutionary multiobjective optimization,” IEEE Trans. Evol. Comput., vol. 18, no. 6, pp. 909–923, Dec. 2014.
[10] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, “A reference vector guided evolutionary algorithm for many-objective optimization,” IEEE Trans. Evol. Comput., vol. 20, no. 5, pp. 773–791, Oct. 2016.
[11] G. Tian, Y. Ren, Y. Feng, M. Zhou, H. Zhang, and J. Tan, “Modeling and planning for dual-objective selective disassembly using and or graph and discrete artificial bee colony,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2456–2468, Apr. 2019.

[12] Y. Feng, M. Zhou, G. Tian, Z. Li, Z. Zhang, Q. Zhang, and J. Tan, “Target disassembly sequencing and scheme evaluation for CNC machine tools using improved multiobjective ant colony algorithm and fuzzy integral,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2438–2451, Dec. 2019.

[13] K. Li, K. Deb, Q. Zhang, and S. Kwong, “An evolutionary many-objective optimization algorithm based on dominance and decomposition,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[14] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, “Balancing convergence and diversity in decomposition-based many-objective optimizers,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, Apr. 2016.

[15] X. Chen, D. Zhang, and X. Zeng, “A stable matching-based selection and memory enhanced MOEA/D for evolutionary dynamic multiobjective optimization,” in *Proc. IEEE 27th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2015, pp. 478–485.

[16] L. Cao, L. Xu, E. D. Goodman, and S. Zhu, “Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor,” *IEEE Trans. Evol. Comput.*, to be published, doi: 10.1109/TEVC.2019.2925722.

[17] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, “A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization,” *IEEE Trans. Cybern.*, to be published.

[18] S.-Y. Zeng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, and L. Kang, “A dynamic multi-objective evolutionary algorithm based on an orthogonal design,” in *Proc. IEEE Int. Conf. Evol. Comput.*, Sep. 2006, pp. 573–580.

[19] B. Zheng, “A new dynamic multi-objective optimization evolutionary algorithm,” in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, Aug. 2007, vol. 5, no. 1, pp. 565–570.

[20] J. Zeng, L. Xu, E. D. Goodman, and H. Li, “A first-order differential evolutionary algorithm for flexible manufacturing systems—A survey,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 296–310, Jun. 2007.

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

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

[23] L. Cao, L. Xu, E. D. Goodman, and H. Li, “A first-order difference model-based evolutionary dynamic multiobjective optimization,” in *Proc. SEAL*, in Lecture Notes in Computer Science, vol. 9598, 2016, pp. 644–655.

[24] 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.

[25] K. Deb and S. Karthik, “Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling,” in *Proc. EMO*, in Lecture Notes in Computer Science, vol. 4403, 2007, pp. 803–817.

[26] S.-Y. Zheng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, and L. Kang, “A dynamic multi-objective evolutionary algorithm based on fuzzy integral,” in *Proc. IEEE Int. Conf. Evol. Comput.*, Sep. 2006, pp. 573–580.

[27] B. Zheng, “A new dynamic multi-objective optimization evolutionary algorithm,” in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, Aug. 2007, vol. 5, no. 1, pp. 565–570.

[28] Y. Wang and B. Li, “Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment,” in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 630–637.

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

[30] S. Azzouz, K. Deb, 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, Feb. 2017.

[31] I. Hatzakis and D. Wallace, “Dynamic multi-objective optimization evolutionary algorithms: A forward-looking approach,” in *Proc. GECCO*, vol. 4, 2006, pp. 1201–1208.

[32] A. Zou, Q. Li, Y. Jin, and Q. Zhang, “A population prediction strategy for evolutionary dynamic multi-objective optimization,” *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, Jan. 2014.

[33] X. Chen, D. Zhang, and X. Zeng, “A stable matching-based selection and memory enhanced MOEA/D for evolutionary dynamic multiobjective optimization,” in *Proc. 2015 IEEE Int. Conf. Cybern*. (ICCB), Jun. 2015, pp. 478–485.

[34] S.-Y. Zeng, G. Chen, L. Zheng, H. Shi, H. de Garis, L. Ding, and L. Kang, “A dynamic multi-objective evolutionary algorithm based on fuzzy integral,” in *Proc. IEEE Int. Conf. Evol. Comput.*, Sep. 2006, pp. 573–580.

[35] B. Zheng, “A new dynamic multi-objective optimization evolutionary algorithm,” in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, Aug. 2007, vol. 5, no. 1, pp. 565–570.

[36] Y. Wang and B. Li, “Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment,” in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 630–637.
CHANGGAO XIA received the B.S., M.S., and Ph.D. degrees from the School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang, China. He is currently a Professor with the School of Automotive and Traffic Engineering, Jiangsu University. His research interest includes modeling and control of vehicle dynamics.

LONG CHEN received the Ph.D. degree from Jiangsu University, Zhenjiang, China, in 2006. He is currently a Professor with the School of Automotive and Traffic Engineering, Jiangsu University. His main research interest includes modeling and control of vehicle dynamics.

YINGFENG CAI received the B.S., M.S., and Ph.D. degrees from the School of Instrument Science and Engineering, Southeast University, Nanjing, China. In 2013, she joined the Automotive Engineering Research Institute, Jiangsu University, as an Assistant Professor, where she has been a Professor since 2018. She is currently the Director of the Automotive Engineering Research Institute. Her research interests include computer vision, intelligent transportation systems, and intelligent automobiles.

* * *