# A Dynamic Multi-objective Evolutionary Algorithm Based on Decision Variable Classification

Zhengping Liang, Tiancheng Wu, Xiaoliang Ma, Zexuan Zhu and Shengxiang Yang

***Abstract*—In recent years, dynamic multi-objective optimization problems (DMOPs) have drawn increasing interest. Many dynamic multi-objective evolutionary algorithms (DMOEAs) have been put forward to solve DMOPs mainly by incorporating diversity introduction or prediction approaches with conventional multi-objective evolutionary algorithms. Maintaining good balance of population diversity and convergence is critical to the performance of DMOEAs. To address the above issue, a dynamic multi-objective evolutionary algorithm based on decision variable classification (DMOEA-DVC) is proposed in this study. DMOEA-DVC divides the decision variables into two and three different groups in static optimization and change response stages, respectively. In static optimization, two different crossover operators are used for the two decision variable groups to accelerate the convergence while maintaining good diversity. In change response, DMOEA-DVC reinitializes the three decision variable groups by maintenance, prediction, and diversity introduction strategies, respectively. DMOEA-DVC is compared with the other six state-of-the-art DMOEAs on 33 benchmark DMOPs. Experimental results demonstrate that the overall performance of the DMOEA-DVC is superior or comparable to that of the compared algorithms.**

***Index Terms*—Dynamic multi-objective optimization problem, multi-objective optimization problem, dynamic multi-objective evolutionary algorithm, multi-objective evolutionary algorithm, decision variable classification.**

## I. Introduction

Dynamic multi-objective optimization problems (DMOPs), with multiple conflicting and time-varying objectives, are ubiquitous in real-world applications [1], [2], [3], [4], [5], [6]. Multi-objective evolutionary algorithms (MOEAs) [7], [8], [9], [10], [11], [12], [13], [14] have achieved success on various static multi-objective optimization problems (MOPs) [15], [16], [17], [18]. However, they tend to fail in DMOPs due to the lack of quick change response mechanism in dynamic environment. To solve DMOPs, many dynamic MOEAs (DMOEAs) have been proposed in recent years. The majority of them can be categorized into diversity introduction approaches [1], [19], [20], [21], [22], [23], [24] and prediction approaches [25], [26], [27], [28], [29], [30], [31], [32], [33].

Diversity introduction approaches introduce a certain proportion of randomized or mutated individuals into the evolution population once a change occurs to increase the population diversity. The increase of diversity can facilitate the algorithms to better adapt to the new environment. However, since these algorithms mainly rely on the static evolution search to find the optimal solution set after diversity introduction, the convergence might be slow down. Prediction approaches adopt prediction models to predict the promising population in the changing environments. They can substantially improve the convergence of the population. However, most prediction models require a training cycle in which the performance of the prediction models is unsatisfactory. Moreover, the existing DMOEAs, including the diversity introduction and prediction approaches as well as other miscellaneous approaches [34], [35], [36], [37], [38], [39], do not take into account the different characteristics of the decision variables. They tend to explore all decision variables in the same way, which is less efficient in balancing the population diversity and convergence.

To address the aforementioned issues, in this work, we propose a DMOEA based on decision variable classification (DMOEA-DVC). DMOEA-DVC is characterized by a combination of diversity introduction, fast prediction models, and decision variable classification methods. The diversity introduction and prediction methods complement with each other in DMOEA-DVC to overcome their inherent defects. Two decision variable classification methods are introduced to classify the decision variables into two and three different groups in static optimization and change response stages, respectively. Based on the classification, different evolution operators and change response strategies are accordingly applied on the decision variables to enhance the population diversity and convergence. DMOEA-DVC is compared with the other six state-of-the-art DMOEAs including DNSGA-II-B [1], PPS [25], MOEA/D-KF [26], SGEA [33], Tr-DMOEA [35], and DMOEA-CO [52] on 33 benchmark DMOPs, including five FDA benchmarks [4], three dMOP benchmarks [19], two DIMP benchmarks [41], nine JY benchmarks [42], and fourteen newly developed DF benchmarks [43]. The experimental results show that DMOEA-DVC is more adaptable to the

Manuscript received XX. XX. XXX; revised XX. XX. XXX; accepted XX. XX. XXX. This work was supported in part by the National Natural Science Foundation of China under Grants 61871272, 61976143 and 61673331, in part by the Natural Science Foundation of Guangdong Province, China, under Grants 2020A151501479, 2019A1515010869 and 2020A151501946, in part by the Shenzhen Scientific Research and Development Funding Program under Grants JCYJ20190808173617147 and GGFW2018020518310863, in part by the Scientific Research Foundation of Shenzhen University for newly introduced teachers, under Grant 2019048 and 85304/00000247, and in part by the Zhejiang Lab's International Talent Fund for Young Professionals.
*(Corresponding author: Zexuan Zhu.)*

Z. Liang, T. Wu, and X. Ma are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: liangzp@szu.edu.cn; wutianchengsz@163.com; maxiaoliang@yeah.net;).

Z. Zhu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, also with Shenzhen Pengcheng Laboratory, Shenzhen 518055, China, and also with the SZU Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen University, Shenzhen 518060, China (e-mail: zhuzx@szu.edu.cn).

S. Yang is with the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (email: syang@dmu.ac.uk).

changing environments than the other six algorithms.

The contributions of this work are summarized as follows:

1) Two new decision variable classification methods, applicable to the static optimization and change response stages, respectively, enable the algorithm to explore different variables more efficiently.

2) In static optimization, a new offspring generation strategy by mixing specific crossover operators for two different kinds of decision variables is introduced to speed up the population convergence, while maintaining the population diversity of the algorithm.

3) In change response, a hybrid response strategy of maintenance, prediction, and diversity introduction is advanced to handle three types of decision variables, such that better adaptability in different dynamic environments is achieved.

The rest of this paper is organized as follows: Section II introduces the basics of DMOP and the related work of the existing DMOEAs, Section III describes the details of the proposed DMOEA-DVC, Sections IV and V present the experimental design and results, respectively, and finally Section VI concludes this study and discusses the future work.

## II. Background and Related Work

This section provides some basics of DMOP to facilitate the understanding of the proposed method and reviews the related work on DMOEAs and decision variable classification.

### A. Basics of DMOP

Without loss of generality, we can assume that each objective function of a DMOP is a minimization problem, then a DMOP can be mathematically defined as follows:

$$\min_{x\in\Omega} \mathbf{F}(x,t) = (f_1(x,t),...,f_m(x,t))^T \tag{1}$$

where $x = (x_1, x_2,..., x_n)^{\mathrm{T}}$ is an $n$-dimensional decision vector bounded in the decision space $\Omega$, $m$ is the number of objectives, $t$ is the time parameter, and the mapping function $f_i(x, t)$ ($i$ = 1, 2, …, $m$) refers to the $i$-th objective function of $x$ at time $t$. $\mathbf{F}(x, t)$ is the objective function vector that evaluates solution $x$ at time $t$. The mathematical definition of time parameter $t$ is given as follows:

$$t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor \tag{2}$$

where $\tau$ is the iteration counter, $n_t$ is the number of distinct steps in $t$, and $\tau_t$ is the number of iterations for which $t$ remains the same. In Eq. (2), $n_t$ and $\tau_t$ determine the severity level and frequency value of $t$, respectively.

The goal of solving Eq. (1) is to find the dynamic Pareto-optimal set or dynamic Pareto-optimal front based on dynamic Pareto-dominance:

***Definition 1 Dynamic Pareto-dominance***: Given two candidate solutions $x$ and $y$ ($x, y \in \Omega$) at time $t$, $x$ is said to dominate $y$, written as $x(t) \prec y(t)$, if and only if

$$\begin{aligned}&(\forall i \in \{1,2,...,m\}: f_i(x,t) \le f_i(y,t))\\ &\qquad \wedge (\exists j \in \{1,2,...,m\}: f_j(x,t) < f_j(y,t))\end{aligned} \tag{3}$$

***Definition 2 Dynamic Pareto-optimal set***: A dynamic Pareto-optimal set at time $t$, denoted as $PS(t)$, includes all solutions that are not dominated by any other solutions at time $t$, i.e.,

$$PS(t) = \{x \mid \neg\exists y \in \Omega : x(t) \prec y(t)\} \tag{4}$$

***Definition 3 Dynamic Pareto-optimal front***: A dynamic Pareto-optimal front at time $t$, denoted as $PF(t)$, is the mapping of the solutions in $PS(t)$ in the objective space, i.e.,

$$PF(t) = \{\mathbf{F}(x,t) \mid x \in PS(t)\} \tag{5}$$

***Definition 4 Multi-optimal variable***: Given a decision variable $i$, if there exist two solutions $x$ and $y$ in $PS(t)$ and $x_i \neq y_i$, where $x_i$ and $y_i$ are the $i$-th decision variable values of candidate solutions $x$ and $y$, respectively, then the decision variable $i$ is said to have multiple optimal values, i.e., it is a multi-optimal variable.

***Definition 5* Single-optimal variable:** Given a decision variable $i$, if for any two solutions $x$ and $y$ in $PS(t)$, their corresponding $i$-th decision variable values are the same, i.e., $x_i = y_i$, then the decision variable $i$ is said to have single optimal value, i.e., it is a single-optimal variable.

In other words, if a decision variable is single-optimal, all solutions in $PS(t)$ have the same value on this decision variable. If a decision variable is multi-optimal, there have multiples values for this variable among the individuals in $PS(t)$.

In general, a DMOP can be divided into four types [4] according to the dynamic characteristics of $PF(t)$ and $PS(t)$:

Type I: $PS(t)$ changes over time and $PF(t)$ is fixed.

Type II: Both $PS(t)$ and $PF(t)$ change over time.

Type III: $PS(t)$ is fixed and $PF(t)$ changes over time.

Type IV: Both $PS(t)$ and $PF(t)$ are fixed, but the problem changes over time.

### B. Dynamic Multi-Objective Evolutionary Algorithms

In recent years, many DMOEAs have been put forward to deal with DMOPs. These algorithms are mainly categorized into two groups, i.e., diversity introduction and prediction based approaches.

Diversity introduction approaches take into account the potential diversity loss of the population in a dynamic environment, and introduce randomized or mutated individuals once an environmental change is detected. For example, Deb et al. [1] proposed two DMOEAs (DNSGA-II-A and DNSGA-II-B) based on NSGA-II [7]. Once a change is detected, DNSGA-II-A randomly re-initializes 20% of the individuals, while DNSGA-II-B randomly mutates 20% of the individuals. Goh and Tan [19] introduced a competitive-cooperative coevolutionary algorithm (dCOEA) where some new individuals are generated randomly to enhance the diversity of the population when the environmental changes. Helbig and Engelbrecht [20] proposed a heterogeneous dynamic vector evaluated particle swarm optimisation (HDVEPSO) algorithm by combining heterogeneous particle swarm optimisation (HPSO) [21], [22] and dynamic vector evaluated particle swarm optimisation (DVEPSO) [23]. HDVEPSO randomly re-initializes 30% of the swarm particles after the objective function changes. Martínez-Peñaloza and Mezura-Montes [24] combined generalized differential evolution along with an artificial immune system to solve DMOP (Immune-GDE3). As demonstrated in the existing studies, diversity introduction approaches can prevent the population from falling into local optima and they are easy to implement.

To accelerate the adaptation of the population to the dynamic environments, prediction based approaches have been proposed to generate a promising population in a new environment. For example, Zhou et al. [25] presented a population prediction strategy (PPS) to divide the population into a center point and a manifold. The proposed method uses an autoregression (AR) model to locate the next center point, and uses the previous two consecutive manifolds to predict the next manifold. The predicted center point and manifold make up a new population more suitable to the new environment. Muruganantham et al. [26] applied Kalman filter [44] in the decision space to predict the new Pareto-optimal set. They also proposed a scoring scheme to decide the predicting proportion. Hatzakis and Wallace [27] suggested an AR model to predict the boundary point in the objective space in the new environment. Peng et al. [28] put forward exploration and exploitation operators to predict the new optimal solutions. Wei and Wang [29] presented hyper rectangle prediction to generate a solution set once a change occurs. Ruan et al. [30] applied gradual search to predict the ideal position of the individuals in the new environment. Wu et al. [31] re-initialized individuals in the orthogonal direction to the predicted direction of the population in change response. Ma et al. [32] utilized a simple linear model to generate the population in the new environment. Jiang and Yang [33] introduced a steady-state and generational evolutionary algorithm (SGEA), which guides the search of the solutions by a moving direction from the centroid of the non-dominated solution set to the centroid of the whole population. The step-size of the search is defined as the Euclidean distance between the centroids of the non-dominated solution set at time steps ($t$-1) and $t$. In the aforementioned references, prediction based approaches have shown capabilities of enhancing the convergence speed.

This work proposes an enhance change response strategy by combining diversity introduction with a fast prediction based approach to take the advantages of both.

### C. Decision Variable Classification Methods

A diversity introduction or prediction approach can be regarded as a probabilistic model for searching the optimal values of the decision variables. Most existing DMOEAs assume all decision variables are under the same probability distribution. However, in real DMOPs, the probability distributions of the decision variables can be significantly varied. With decision variable classification, decision variables can be classified into different groups, and then specific probabilistic searching models can be applied to the corresponding variable groups to obtain better solutions.

Many MOEAs based on decision variable classification have achieved success on static MOPs. For example, decision variable classification is implemented by decision variable perturbation in [45], [46], [47], [48]. Decision variable perturbation generates a large number of individuals for classification and consumes proportionally a large number of fitness evaluations. This strategy works well for static MOPs where the categories of decision variables do not change, and it calls for only once classification. However, in DMOPs, the categories of decision variables may change over time and much more times of classification and fitness evaluations are required. Sun et al. [49] and Omidvar et al. [50] reduced the number of fitness evaluations by using statistical information collected over a certain period of time. Unfortunately, the environment could change quickly in DMOPs, which does not allow such methods to obtain accurate statistical information in the short time windows.

There are few methods proposed for decision variable classification in DMOPs and the existing methods for static problems might not be applicable to dynamic problems. Woldesenbet and Yen [51] distinguished the decision variables by their average sensitivities to the change in the objective space, based on which individuals are relocated. The method works well for dynamic single-objective optimization problems, but it is inapplicable to DMOPs. Xu et al. [52] introduced a cooperative co-evolutionary algorithm for DMOPs where the decision variables are decomposed into two subcomponents, i.e., inseparable and separable variables with respect to the environment variable $t$. Two populations are applied to cooperatively optimize the two subcomponents, respectively. The proposed algorithm in [52] is superior on DMOPs where the decision variables are decomposable based on environment sensitivities, however, it might not be the fact in many DMOPs.

In this work, we propose a more general decision variable classification method applicable to most DMOPs. The proposed method achieves accurate classification with no extra objective function evaluations nor iterative accumulation to collect statistical information. Particularly, the decision variable classification method uses the statistical information between the decision variables and the objective functions that is available in the first iteration after each environment change, i.e., no need to consume extra fitness evaluations. It is worth highlighting that the classification proposed in this work is the first attempt to distinguish the decision variable distributions (i.e., single-optimal or multi-optimal value) in DMOPs. From the beginning of the search, different strategies are adopted to sample different decision variables. In this way, decision variables can obey the distribution of $PS(t)$ as much as possible in the iterative process, so as to better cover and approach $PS(t)$.

## III. The Proposed Framework and Implementation

DMOEA-DVC uses a steady-state and generational MOEA as the static framework, i.e., DMOEA-DVC responds to the changes and generates individuals in a steady-state manner and performs environmental selection in a generational manner.

**Algorithm 1:** Framework of DMOEA-DVC

1 **Input**: $N$ (population size)
2 **Output**: $P$ (parent population)
3 Initial parent population $P = \{x^1, x^2, ..., x^N\}$;
4 $(A, P') =$ EnvironmentSelection($P$);
5 **while** *stopping criterion not met*
6 *flag_multi* = ClassificationSO ($P$);
7 **for** $i = 1$ to $N$
8 **if** *change detected and not responded*
9 ($P$, $A$, $P'$, *flag_multi*) = ChangeResponse($P$);
10 **end if**
11 $x$ = GenerateOffspring(*flag_multi*, $P$, $A$);
12 ($P$, $A$) = UpdatePopulation($x$);
13 **end for**
14 ($A$, $P$) = EnvironmentSelection($P \cup P'$);
15 **end while**

**Algorithm 2:** ClassificationSO($P$) //SO: Static Optimization

**Input**: $P$ (parent population)
**Output**: *flag_multi* (each item is a Boolean value indicating whether the corresponding variable is multi-optimal variable)
**for** $i = 1$ to $n$ // $n$ refers to the number of decision variables
**for** $j = 1$ to $m$ // $m$ refers to the number of objective functions
Compute $r_i^j(t)$ according to Eq. (6);
**end for**
**end for**
**for** $i = 1$ to $n$
**if** $\max(r_i^1(t), r_i^2(t),\ldots,r_i^m(t))>0.5\alpha$ & $\min(r_i^1(t), r_i^2(t),\ldots, r_i^m(t))<-0.5\alpha$
*flag_multi*[$i$] = **true**;
**else**
*flag_multi*[$i$] = **false**;
**end if**
**end for**

The overall framework of DMOEA-DVC is shown in Algorithm 1

At the beginning of the algorithm, a parent population $P$ is initialized. An offspring population $P'$ and an archive $A$ of non-dominated solutions are then selected from the parent population. In each evolution generation, before each individual is generated using evolution operators, the decision variable classification is applied on $P$ and the result is recorded in a Boolean vector *flag_multi*, of which each element indicates whether a corresponding decision variable is a multi-optimal or single-optimal variable. DMOEA-DVC detects any potential changes during the evolution. If a change is detected, the change response strategy is applied. Otherwise, it uses different crossover operators to generate the values of different types of variables in each individual. Once an individual is generated, both $P$ and $A$ are updated. At the end of each generation, a new offspring population and a new archive is selected from the combination of the current parent population and offspring population.

The originality of this work lies in the decision variable classification (lines 6 and 9 of Algorithm 1), offspring generation (line 11 of Algorithm 1), and change response strategy (line 9 of Algorithm 1). The key components of DMOEA-DVC are described in details in the following subsections.

### *A. Decision Variable Classification*

*1) Decision Variable Classification in Static Optimization:* Decision variable classification in static optimization stage (line 6 of Algorithm 1) is applied to increase the possibility of generating high quality offspring individuals. A good DMOEA should be able to find a population with uniform distribution on the PF and converge to the PF as quickly as possible. To achieve this goal, it is favorable to explore the neighborhood of the non-dominated individuals in the population. However, if all the decision variables of the offspring are generated near that of the non-dominated individuals, the population tends to be trapped in the local optima. To solve this issue, the values of multi-optimal variables in the offspring individuals should be generated far away from that of the parent individuals. The reason is that the optimal values of multi-optimal variables may spread across a specific area. At the early search stage, the non-dominated individuals may not be able to cover this area, therefore the population cannot possess good diversity if the offspring are generated near the non-dominated individuals. In this regard, the offspring should be generated far away from the non-dominated individuals on multi-optimal variables to maintain good diversity. Conversely, for single-optimal variables, the generated values in the offspring individuals should be as close as possible to the corresponding values in the parent individuals to accelerate the convergence.

The key issue is how to classify multi-optimal and single-optimal variables. Since the real optimal solutions of the target problems are usually unknown in advance, it is infeasible to classify the variables based on the definitions provided in Section II.A. In this subsection, we propose an approximate method to distinguish multi-optimal and single-optimal variables. In DMOP, the objective functions could conflict with each other on some decision variables [53], [54]. If two objective functions conflict on a decision variable, the decision variable is deemed to have multiple optimal values.

Given two non-dominant solutions $x$ and $y$, without loss of generality, suppose $x_i > y_i$ on the $i$-th decision variable. If there exist two different objective functions $f_j(\cdot)$ and $f_k(\cdot)$ such that $f_j(x,t) > f_j(y,t)$ and $f_k(x,t) < f_k(y,t)$, $f_j(\cdot)$ is positively correlated with the $i$-th decision variable whereas $f_k(\cdot)$ is negatively correlated with decision variable $i$. Since the monotonicities of $f_j(\cdot)$ and $f_k(\cdot)$ conflict with each other with respect to the $i$-th decision variable, the Pareto optimal solutions are unable to reach a consensus on variable $i$. As such, the $i$-th decision variable is classified as multi-optimal variable. In other cases, it is not straightforward to determine whether the $i$-th decision variable has multiple or single optimal values. For the sake of saving computational budget, the other decision variables are classified as single-optimal variables, which usually holds in real DMOPs. In this line, we use Spearman rank correlation coefficient (SRCC) to measure the correlation between one variable and an objective function. Particularly, the SRCC between a decision variable $i$ ($i = 1,2,...,n$) and an objective function $f_j(x, t)$ ($j = 1,2,...,m$) is defined as follows:

**Algorithm 3:** ClassificationCR(*P*) //CR: Change Response

```
Input: P (parent population)
Output: ttest (result of t-test), flag_predict (each item is a Boolean value indicating whether the corresponding variable is re-initialized by prediction or not), x_center (center individual), x_trial (a set of trial individuals)
Compute ttest values according to Eq. (7);
Generate a center individual x_center and evaluate x_center;
Generate x_p by x_center using a prediction model;
for i = 1 to n
    for j = 1 to n
        if i == j
            x_trial[j]_i = x_p_i;
        else
            x_trial[j]_i = x_center_i;
        end if
    end for
end for
for i = 1 to n
    Evaluate x_trial[i];
    if x_trial[i] ≺ x_center
        flag_predict[i] = true;
    else
        flag_predict[i] = false;
    end if
end for
```

$$r_i^j(t) = 1 - \frac{6\sum_{k=1}^{N}(d_{ij}^k)^2}{N(N^2-1)} \tag{6}$$

where $N$ refers to the size of population. $d_{ij}^k$ represents the difference between the rank of the $i$-th decision variable and the rank of the $j$-th objective value in the $k$-th individual. The rank of a variable is the rank of its ascending order in $P$. The SRCC value $r_i^j(t)$ ranges from -1 to 1. A positive value indicates that the objective function tends to increase as the decision variable increases. In contrast, a negative value suggests that the objective function is likely to decrease as the decision variable increases. If there is an obvious correlation confliction (positive correlation vs. negative correlation) between two objective functions on the $i$-th variable, i.e., $\max(r_i^1(t), r_i^2(t),\ldots, r_i^m(t)) > 0.5\alpha$ and $\min(r_i^1(t), r_i^2(t),\ldots, r_i^m(t)) < -0.5\alpha$ ($\alpha$ is a predefined threshold), then the $i$-th decision variable is classified as a multi-optimal variable. Otherwise, the $i$-th decision variable is considered as a single-optimal variable.

The implementation of the decision variable classification strategy in static optimization is presented in Algorithm 2. Firstly, the SRCC value between each decision variable $i$ and each objective $j$, i.e., $r_i^j$, is calculated (lines 3 to 7). Secondly, the decision variable is classified as multi-optimal variable (line 10) or single-optimal variable (line 12) according to the SRCC values.

*2) Decision Variable Classification in Change Response:* In most existing DMOPs, the decision variables can be categorized into similar, predictable, and unpredictable variables, considering the environment changes. A similar variable is a variable that hardly varies in the last two consecutive environment changes. A predictable variable denotes a variable on which prediction achieves significant performance improvement in the previous two environment changes. Conversely, an unpredictable variable has no gain in prediction. Intuitively, the similar decision variables need no re-initialization in the change response; the predictable decision variables should be re-initialized based on prediction, and the unpredictable decision variables can be re-initialized via diversity introduction.

The nonparametric t-test [55], [56] is used to evaluate the correlation of the change of a decision variable and the environment changes. Particularly, the values of a decision variable $i$ in the current population and the previous environment are tested as follows:

$$\text{ttest}_i = \frac{\left|\bar{x}_i(t) - \bar{x}_i(t-1)\right|}{\sqrt{\dfrac{(\text{Var}(x_i(t)))^2 + (\text{Var}(x_i(t-1)))^2}{N}}} \tag{7}$$

where $\bar{x}_i(t)$ represents the average value of the variable $i$ in population $P$ at time step $t$. If $\text{ttest}_i \le \beta$, where $\beta$ is a predefined threshold, the $i$-th variable is considered subject to insignificant change, i.e., it is a similar variable and no re-initialization is required. If $\text{ttest}_i > \beta$, then the $i$-th decision variable is deemed to have significant change, i.e., variable $i$ is predictable or unpredictable and re-initialization is needed. An extra procedure is carried out to determine whether variable $i$ is predictable or unpredictable:

Firstly, a centroid individual of the current population is generated as follows:

$$x\_center = \frac{1}{|P|}\sum_{x \in P} x \tag{8}$$

Thereafter, $n$ trial individuals $x\_trial[i]$ ($i$=1, 2,…, $n$) are generated by placing the $i$-th decision variable of $x\_center$ with a predicted value while leaving the other decision variables unchanged. The prediction can be done with any reasonable model (described in next subsection). If $x\_trial[i]$ dominates $x\_center$, the prediction of the $i$-th decision variable is accepted, and the $i$-th decision variable is classified as a predictable variable and re-initialization based on prediction is applied. Otherwise, the $i$-th decision variable is unpredictable and re-initialization based on diversity introduction is used. Note that the centroid individual and the trial individuals are added to the current population $P$, so that the decision variable classification in change response does not cause extra evaluation.

Theoretically, a prediction model should also be able to predict the situation where decision variables do not change too much in different environments. However, in some DMOPs (like FDA2, dMOP1, JY5, and JY8 used in Section IV), the $PS(t)$ is unchanged, and a tiny change in a decision variable value can lead to significant negative impact on the algorithms performance. Prediction models are not applicable to such problems.

The procedure of decision variable classification in change response is outlined in Algorithm 3. Firstly, the ttest values and $x\_center$ are calculated according to Eqs. (7) and (8), respectively. Secondly, a prediction model is used to predict a new individual $x\_p$ (the exact prediction model is described in

**Algorithm 4:** ChangeResponse(*P*)

```
1  Input: P (parent population)
2  Output: P (updated parent population), A (updated archive
   population), P′ (updated offspring population), flag_multi
3  (ttest, flag_predict, x_center, x_trial) = ClassificationCR(P);
4  P′ = x_trial ∪ {x_center};
5  Clear A and copy nondominated solutions in P′ to A;
6  for i = 1 to m
7      y = argmin(f_i(x^1, t), f_i(x^2, t), …, f_i(x^N, t));
8      Evaluate individual y and add y to P′;
9      Remove all solutions in A that is dominated by y;
10     Add y to A if y is not dominated by any other solutions in A;
11 end for
12 for i = 1 to N - m - n -1
13     for j = 1 to n
14         if ttest[j] > β
15             if flag_predict[j] == false;
16                 Re-initialize x_j^i by Eq. (10);
17             Else
18                 Re-initialize x_j^i by Eq. (12);
19             end if
20         end if
21     end for
22     Evaluate individual x^i and add x^i to P′;
23     Remove all solutions in A that is dominated by x^i;
24     Add x^i to A if x^i is not dominated by any other solutions in A;
25 end for
26 P = P′;
27 flag_multi = ClassificationSO (A);
```

Section IV.C). Thirdly, trial individuals are generated (lines 6 to 14) and evaluated. If a trial individual dominates the centroid individual, the prediction of the corresponding decision variable of the trial individual is correct, otherwise the prediction is unacceptable.

### B. Environmental Selection

DMOEA-DVC uses the same environmental selection strategy as SGEA [33]. The environmental selection process starts with fitness assignment. Each individual $x^i$ of the current population $P$ is assigned a fitness value $F(i)$, which is defined as the number of individuals dominating $x^i$, i.e.,

$$F(i) = \left|\{x^j(t) \in P \mid x^j(t) \prec x^i(t)\}\right| \tag{9}$$

where $|\cdot|$ denotes the cardinality of a set. If $F(i)$=0, $x^i$ is non-dominated by any other individuals, all non-dominated individuals are then copied to the archive $A$. If $|A|$ is smaller than the population size $N$, the best $N$ individuals are preserved in the offspring population $P'$ based on their fitness values. If $|A|$ is equal to $N$, all non-dominated individuals are copied to $P'$. If $|A|$ is greater than $N$, the farthest first method [57], [58] is used to select $N$ individuals from $A$ to $P'$.

### C. Change Response

The overall procedure of the change response strategy in DMOEA-DVC is presented in Algorithm 4. In change response, DMOEA-DVC applies maintenance, diversity introduction, and prediction approach for the three types of decision variables defined in Section III.A, respectively:

1) *Maintenance:* If a decision variable is a similar variable, DMOEA-DVC leaves the value of this variable unchanged in change response.

**Algorithm 5:** GenerateOffspring(*flag_multi*, *P*, *A*)

```
Input: flag_multi (set of Boolean values to indicate whether a
variable is multi-optimal variable or not), P (parent population), A
(archive population)
Output: x (offspring individual)
Random pick an individual a from archive A;
Perform binary tournament selection on P to select two distinct
individuals b and c;
for i = 1 to n
    if (flag_multi[i] == true)
        x_i = DE(a_i, b_i, c_i);
    else
        x_i = SBX(a_i, b_i);
    end if
end for
PolynomialMutation(x);
evaluate x;
```

2) *Diversity introduction:* If a decision variable is an unpredictable variable, a random re-initialization strategy is applied on this variable. Particularly, the variable is updated as follows:

$$x_i(t+1) = \mathrm{L}_i(t) + rand * (\mathrm{U}_i(t) - \mathrm{L}_i(t)) \tag{10}$$

where $\mathrm{L}_i(t)$ and $\mathrm{U}_i(t)$ represent the lower and upper bounds of the $i$-th variable at time step $t$, respectively, and *rand* is a random value in [0,1].

3) *Prediction approach:* If a decision variable is classified as predictable, it is re-initialized by the center prediction with Kalman filter [25], [33] that is characterized by short training cycle. Kalman filter provides an efficient computational means to estimate the state of a process, in a way that minimizes the mean of the squared error [59]. Here we utilize Kalman filter to predict the population center $x\_p$ at time step ($t$+1), i.e.,

$$x\_p = \mathrm{Kalman}(x\_center(t)) \tag{11}$$

and use the predicted center to re-initialize the predictable decision variables as follows:

$$x_i(t+1) = x_i(t) + x\_p_i - x\_center_i(t) \tag{12}$$

where $x\_p_i$ is the $i$-th decision variable of $x\_p$.

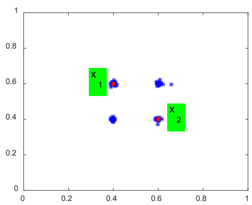
Every new decision variable should be repaired if it violates the boundary constriction, i.e.,

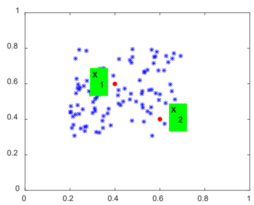
$$x_i(t) = \begin{cases} \mathrm{L}_i(t) & \text{if } x_i(t) < \mathrm{L}_i(t) \\ \mathrm{U}_i(t) & \text{if } x_i(t) > \mathrm{U}_i(t) \end{cases} \tag{13}$$

The boundary points are critical to explore the Pareto front [60], [61]. To prevent from losing the boundary points in the new environment, DMOEA-DVC retains the $m$ boundary points of the population at time step $t$ (lines 6-line 11 of Algorithm 4). On $t$=1, DMOEA-DVC re-initializes all variables by polynomial mutation. At the end of the change response, the decision variable classification in static optimization is implemented because the types of some decision variables in the static optimization may change in different environments.

### D. Offspring Generation

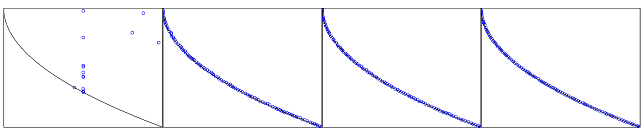
The pseudo-code of offspring generation is shown in Algorithm 5. The simulated binary crossover (SBX) or differential evolution (DE) crossover operator is used to generate the values of a decision variable depending on whether the decision variable is multi-optimal variable or not. SBX and DE crossover



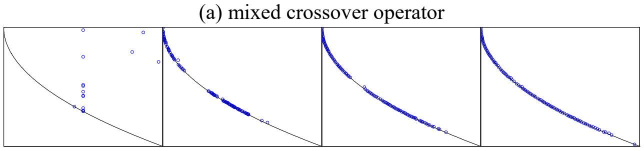


(a) 100 solutions produced by SBX (b) 100 solutions produced by DE

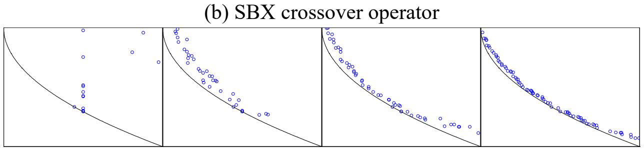
Fig. 1. (a) and (b) show 100 solutions produced by SBX crossover operator and DE crossover operator, respectively. The blue asterisks represent the offspring solutions. The red circle represents the parent solutions. Since DE uses three solutions to generate an offspring solution, the parent solution $x_3$ used by DE is random generated in the whole decision space. The parent solutions $x_1$ and $x_2$ used by SBX and DE are both (0.4, 0.6) and (0.6, 0.4), respectively.



(a) mixed crossover operator



(b) SBX crossover operator



(c) DE crossover operator

Fig. 2. (a), (b), and (c) show the population generated by mixed, SBX, and DE crossover operators in the objective space, respectively. The tested benchmark is FDA1, where the time parameters are all set to zero. The subfigures from left to right are populations in the first, tenth, twentieth, and thirtieth generation, respectively. The black solid line is $PF(0)$.

operators are two commonly used crossover operators in MOEAs. Fig. 1 shows an example of the distribution of the offspring solutions generated by these two crossover operators. The offspring solutions generated by SBX are closer to the parent solutions, whereas the offspring solutions generated by DE crossover are far away from the parent solutions.

As stated in Section III.A, multi-optimal variables should be generated far from the parents, where DE crossover operator should be used. The single-optimal variables should be generated near the parents, i.e., SBX is selected. Fig. 2 shows that the population generated by the mixed crossover operator converges well to $PF(0)$ and achieves an even distribution within the first ten generations. The population generated by the SBX operator can converge to $PF(0)$ within the first ten generations, but it does not show good coverage. The population generated by DE crossover operator is able to achieve a wider distribution within the first twenty generations, but it cannot converge to $PF(0)$. In addition to the mixed crossover operator, DMOEA-DVC also uses polynomial mutation to mutate the solutions.

### *E. Population Update*

DMOEA-DVC uses the same steady population update strategy as SGEA [33]. The population update strategy is carried out in both parent population $P$ and the archive $A$. A newly generated individual $y$ is used to replace the worst individual in $P$, and update $A$. Firstly, if $y$ is not the same as any individual in $P$, $y$ is compared with each individual in $P$. If $y \prec x^i$, the fitness value of $x^i$ is increased by one. If $x^i \prec y$, the fitness value of $y$ is increased by one. Secondly, the individual in $P$ with the highest fitness value is identified as $z$. If $y$ is not worse than $z$ in terms of fitness value, $y$ replaces $z$ and all individuals dominated by $y$ in $A$ are removed. Finally, $y$ is added to $A$ if $A$ is not full.

### *F. Computational Complexity*

The computational cost of the static optimization is mainly involved in the offspring generation, population update, environmental selection, and decision variable classification. The computational complexity of offspring generation is O($mn$), where $m$ is the number of objectives and $n$ is the number of decision variables. The computational cost of population update is O($mN$), where $N$ is the population size. The environmental selection consumes max(O($mN^2$), O($N^2\log N$)) computational cost. In decision variable classification, calculating the rank of each decision variable calls for O($nN^2$) computational cost. Calculating the rank for each objective value and the correlation coefficient matrix require costs of O($mN^2$) and O($nmN$), respectively. As such, the computational cost of decision variable classification in static optimization is max(O($nN^2$), O($mN^2$), O($nmN$)). Overall, the computational cost of DMOEA-DVC for one generational cycle in static optimization is max(O($nN^2$), O($mN^2$), O($nmN$), O($N^2\log N$)). The computational cost of change response is mainly consumed by the archive update, offspring update, and decision variable classification. The computational cost of archive update and offspring population update is O($N^2$). In decision variable classification, the t-test, *x_center*, and *x_trial* are computed at the costs of O($nN$), O($n$), and O($n^2$), respectively. Therefore, the computational complexity of change response is max(O($n^2$), O($N^2$)).

## IV. Experimental Study

The experiment study of this work is designed following [25], [26], and [33]. The proposed DMOEA-DVC is compared with the other six state-of-the-art DMOEAs on 33 benchmark DMOPs. The detailed experimental design and results are presented as follows.

### *A. Benchmark Problems*

A total of 33 benchmarks are tested in this article, including five FDA [4], three dMOP [19], two DIMP [41], nine JY [42], and fourteen DF [43] test problems. Particularly, four of these problems are revised as follows. In FDA2, there is no such characteristic that the $PF$(t) changes from concave to convex as the original paper described. Therefore, to enable the $PF(t)$ of

FDA2 to possess this characteristic, some factors in the objective functions in FDA2 are changed. In the original dMOP3, the values of distance-related variables can be negative, but not the values of position-related variables. In addition, the distance-related and position-related variables may shift one another. These phenomena impede DMOEAs to work on dMOP3. Hence, the ranges of distance-related and position-related variables are set to the same, and the objective function value is always adjusted to be positive. In the original JY4 problem, the objective value may be negative, which is not consistent with the description in the original paper. To fix this issue, we set the negative values to zeros. When solving the original JY9 problem, none of the algorithms can converge to the *PS*(t) at a certain time steps because the *PS*(t) is not in the decision space. In this study, the decision space of JY9 is changed. The details of the revised FDA2, dMOP3, JY4 and JY9 are provided in Table SI of the Supplementary Materials.

### *B. Compared Algorithms and Parameter Settings*

DMOEA-DVC is compared with six state-of-the-art DMOEAs including DNSGA-II-B [1], PPS [25], MOEA/D-KF [26], SGEA [33], Tr-DMOEA [35], and DMOEA-CO [52] in this study. For a fair comparison, most parameters of these algorithms are set according to the original references. Nevertheless, some parameters are tuned to better fit the experiment. The parameter settings are summarized as follows (without specification, the parameter settings used below are applicable to all algorithms):

1) On two-objective and three-objective problems, the population size is set to 100 and 105, respectively. For all problems, the number of decision variables is set to 11.

2) Some algorithms treat change response as an independent generation, but not in the others. For the sake of simplicity, we do not treat change response as an independent generation for all algorithms in this study.

3) It is worth noting that, in the benchmark DMOPs, no change takes place in the first 50 generations, so as to minimize the effect of static optimization [33]. The total number of generations is set to $10n_t\tau_t+50$, i.e., ensuring $10n_t$ changes in all. $n_t$ is fixed to 10 and $\tau_t$ is set to 5, 10 and 20.

4) The parameters $\alpha$ and $\beta$ used in DMOEA-DVC is empirically set to 0.6 and 5, respectively.

5) Each algorithm is run 30 times independently on each test instance and the average results are reported.

### *C. Performance Evaluation Metric*

In this study, the following performance metrics are adopted to evaluated he performance of the algorithms:

*1) Inverted Generational Distance (IGD):* IGD [3], [25], [60] is used as the performance evaluation metric. It evaluates the performance of the algorithms comprehensively in terms of convergence and diversity.

Given the final population obtained by an algorithm $P^*$ and a set of uniformly sampling points *S* in *PF*, the *IGD* is calculated as follows:

$$IGD = \frac{\sum_{i=1}^{|P^*|} \mathrm{d}(P_i^*, S)}{|P^*|} \tag{14}$$

where $|P^*|$ is the cardinality of $P^*$ and $\mathrm{d}(P_i^*, S)$ represents the minimum Euclidean distance in the objective space between the *i*-th point in $P^*$ and *S*. Lower *IGD* value indicates better performance of an algorithm.

*2) Hypervolume (HV):* HV [33] measures the comprehensive performance of an algorithm. It is defined as the *m*-dimensional volume of the region enclosed by the obtained *PF* and a dominated reference point $z_{ref}$ in the objective space.

Give the obtained solutions of an algorithm $P^*$ and a reference point in the objective space $z_{ref}$, the HV indicator measures the space covered by each point of $P^*$ in the objective space. HV corresponds to the nonoverlapping volume of all the hypercubes formed by the reference point $z_{ref}$ and every point in $P^*$, i.e.,

$$HV = \bigcup_{i=1}^{P^*} \{volume_i \mid point_i \in PF\} \tag{15}$$

where $point_i$ is a non-dominated solution in $P^*$, and $volume_i$ is the volume of the hypercube formed by $z_{ref}$ and $point_i$.

To evaluate the performance of an algorithm in a dynamic environment, the *IGD* and *HV* used in this article are actually the average *IGD* and *HV* values of the last generation at each time step. The means and the standard deviations of *IGD* and *HV* value are reported, where the best values among the seven algorithms on each test problem are highlighted in bold type. The Wilcoxon rank-sum test [62] is used to point out the significance between different results at the 0.05 significance level. (Due to space limit, the results of *HV* are provided in Section G of the Supplementary Materials).

### *D. Results on Classic DMOPs*

This subsection presents the *IDG* results of the algorithms on the classic DMOPs namely the FDA, dMOP, DIMP, and JY problems. Due to space constraints, some of the results on JY are provided in Table SII of the Supplementary Materials. From Table I, it is seen that DMOEA-DVC obtains better performance on most of the DMOPs in terms of *IGD* value. However, on FDA2, it performs worse than SGEA. The reason is that the *PS*(*t*) of FDA2 is unchanged, and all decision variable change slightly in different environments. DMOEA-DVC uses DE crossover operator to generate the values of multi-optimal variables, so the values of offspring variables change significantly, which wastes computational resources in searching a wide space. In comparison, SGEA uses SBX crossover operator to generate all the variables, which can exploit a more promising space with fewer computing consumption. On FDA4, Tr-DMOEA is better than DMOEA-DVC, because the *PF*(*t*) of FDA4 does not change, and this type of DMOP is easy for Tr-DMOEA to build model and have a good performance. It is worth noting that, on JY3, all the decision variables are multi-optimal variables, and the decision variable classification in static optimization is actually invalid. However,

TABLE I
THE *IGD* COMPARISON RESULTS ON CLASSIC DMOPS

| Prob. | ($\tau_t$,$n_t$) | DMOEA-DVC | SGEA | DMOEA-CO | DNSGA-II-B | MOEA/D-KF | Tr-DMOEA | PPS |
|---|---|---|---|---|---|---|---|---|
| FDA1 | (5,10) | **1.20E-2(1.50E-3)** | 2.76E-2(3.45E-3) | 4.52E-2(5.65E-3) | 1.40E-1(1.75E-2) | 1.88E-2(2.35E-3) | 1.44E-1(1.80E-2) | 4.12E-2(5.15E-3) |
| | (10,10) | **7.11E-3(8.89E-4)** | 1.27E-2(1.59E-3) | 2.19E-2(2.74E-3) | 4.87E-2(6.08E-3) | 1.10E-2(1.37E-3) | 5.65E-2(7.06E-3) | 4.04E-2(5.04E-3) |
| | (20,10) | **5.53E-3(6.92E-4)** | 6.93E-3(8.66E-4) | 1.11E-2(1.39E-3) | 2.32E-2(2.90E-3) | 7.61E-3(9.51E-4) | 3.00E-2(3.75E-3) | 1.23E-2(1.54E-3) |
| FDA2 | (5,10) | 5.02E-3(6.27E-4) | **4.55E-3(5.69E-4)** | 2.95E-2(3.68E-3) | 7.87E-3(9.84E-4) | 1.10E-2(1.37E-3) | 9.88E-2(1.24E-2) | 3.75E-2(4.68E-3) |
| | (10,10) | 4.78E-3(5.98E-4) | **4.28E-3(5.35E-4)** | 1.17E-2(1.47E-3) | 7.16E-3(8.95E-4) | 8.83E-3(1.10E-3) | 5.38E-2(6.73E-3) | 8.70E-3(1.09E-3) |
| | (20,10) | 4.67E-3(5.84E-4) | **4.14E-3(5.17E-4)** | 7.75E-3(9.69E-4) | 6.83E-3(8.54E-4) | 8.97E-3(1.12E-3) | 2.02E-2(2.53E-3) | 6.74E-3(8.43E-4) |
| FDA3 | (5,10) | **4.90E-2(6.12E-3)** | 8.54E-2(1.07E-2) | 1.90E-1(2.38E-2) | 4.46E-1(5.57E-2) | 8.00E-2(1.00E-2) | 1.48E+0(1.85E-1) | 2.33E-1(2.92E-2) |
| | (10,10) | **3.30E-2(4.12E-3)** | 6.98E-2(8.72E-3) | 1.14E-1(1.42E-2) | 1.81E-1(2.26E-2) | 3.87E-2(4.84E-3) | 6.06E-1(7.58E-2) | 1.95E-1(2.43E-2) |
| | (20,10) | **1.97E-2(2.46E-3)** | 6.23E-2(7.79E-3) | 7.22E-2(9.03E-3) | 3.63E-2(4.53E-3) | 2.17E-2(2.71E-3) | 5.95E-1(7.44E-2) | 1.68E-1(2.10E-2) |
| FDA4 | (5,10) | 1.06E-1(1.33E-2) | 3.30E-1(4.12E-2) | 1.38E-1(1.72E-2) | 6.85E-1(8.57E-2) | 1.22E-1(1.53E-2) | **7.29E-2(9.11E-3)** | 1.45E-1(1.81E-2) |
| | (10,10) | 7.42E-2(9.27E-3) | 1.50E-1(1.87E-2) | 1.20E-1(1.50E-2) | 5.58E-1(6.97E-2) | 9.38E-2(1.17E-2) | **7.10E-2(8.88E-3)** | 1.13E-1(1.41E-2) |
| | (20,10) | **6.40E-2(8.00E-3)** | 7.78E-2(9.72E-3) | 1.17E-1(1.46E-2) | 4.28E-1(5.35E-2) | 7.92E-2(9.90E-3) | 7.09E-2(8.86E-3) | 1.09E-1(1.36E-2) |
| FDA5 | (5,10) | **2.89E-1(3.61E-2)** | 4.39E-1(5.49E-2) | 4.23E-1(5.29E-2) | 6.93E-1(8.66E-2) | 2.94E-1(3.67E-2)= | 8.03E-1(1.00E-1) | 5.82E-1(7.28E-2) |
| | (10,10) | **1.37E-1(1.71E-2)** | 2.17E-1(2.71E-2) | 2.96E-1(3.70E-2) | 5.14E-1(6.42E-2) | 2.15E-1(2.69E-2) | 4.15E-1(5.19E-2) | 2.80E-1(3.50E-2) |
| | (20,10) | **1.00E-1(1.25E-2)** | 1.19E-1(1.49E-2) | 2.06E-1(2.58E-2) | 4.39E-1(5.48E-2) | 1.93E-1(2.41E-2) | 1.71E-1(2.14E-2) | 1.86E-1(2.32E-2) |
| dMOP1 | (5,10) | **4.11E-3(5.14E-4)** | 6.92E-3(8.65E-4) | 9.99E-3(1.25E-3) | 6.03E-3(7.53E-4) | 5.36E-3(6.70E-4) | 4.51E+0(5.64E-1) | 1.33E-1(1.66E-2) |
| | (10,10) | **4.03E-3(5.04E-4)** | 4.70E-3(5.88E-4) | 5.71E-3(7.14E-4) | 6.17E-3(7.72E-4) | 4.49E-3(5.61E-4) | 2.34E+0(2.92E-1) | 6.46E-2(8.08E-3) |
| | (20,10) | **3.99E-3(4.98E-4)** | 4.09E-3(5.11E-4) | 5.40E-3(6.75E-4) | 6.14E-3(7.68E-4) | 4.08E-3(5.10E-4) | 8.67E-1(1.08E-1) | 2.70E-2(3.37E-3) |
| dMOP2 | (5,10) | **1.46E-2(1.82E-3)** | 3.44E-2(4.30E-3) | 4.32E-2(5.40E-3) | 2.14E-1(2.68E-2) | 2.97E-2(3.71E-3) | 2.98E+1(3.73E+0) | 4.15E-2(5.19E-3) |
| | (10,10) | **7.76E-3(9.70E-4)** | 1.35E-2(1.69E-3) | 2.24E-2(2.80E-3) | 5.93E-2(7.42E-3) | 1.33E-2(1.66E-3) | 2.49E+1(3.12E+0) | 1.56E-2(1.95E-3) |
| | (20,10) | **5.66E-3(7.08E-4)** | 6.81E-3(8.51E-4) | 1.12E-2(1.39E-3) | 2.02E-2(2.53E-3) | 6.81E-3(8.51E-4) | 2.13E+1(2.66E+0) | 1.27E-2(1.58E-3) |
| dMOP3 | (5,10) | **2.07E-2(2.59E-3)** | 1.32E-1(1.65E-2) | 1.59E-1(1.99E-2) | 3.32E-1(4.15E-2) | 9.79E-2(1.22E-2) | 1.60E+0(2.00E-1) | 1.76E-1(2.20E-2) |
| | (10,10) | **8.79E-3(1.10E-3)** | 6.95E-2(8.69E-3) | 6.15E-2(7.69E-3) | 6.13E-2(7.66E-3) | 3.81E-2(4.76E-3) | 1.59E+0(1.99E-1) | 8.00E-2(1.00E-2) |
| | (20,10) | **5.71E-3(7.14E-4)** | 2.20E-2(2.75E-3) | 2.22E-2(2.78E-3) | 1.62E-2(2.02E-3) | 1.26E-2(1.57E-3) | 1.53E+0(1.91E-1) | 7.74E-2(9.68E-3) |
| DIMP1 | (5,10) | **1.57E-2(1.97E-3)** | 3.78E-2(4.72E-3) | 1.86E-1(2.33E-2) | 3.93E-1(4.91E-2) | 4.22E-2(5.27E-3) | 1.32E+0(1.65E-1) | 2.62E-1(3.28E-2) |
| | (10,10) | **8.11E-3(1.01E-3)** | 1.74E-2(2.18E-3) | 3.16E-2(3.95E-3) | 1.36E-1(1.70E-2) | 1.72E-2(2.14E-3) | 5.81E-1(7.27E-2) | 8.91E-2(1.11E-2) |
| | (20,10) | **6.04E-3(7.55E-4)** | 8.74E-3(1.09E-3) | 1.34E-2(1.68E-3) | 4.72E-2(5.90E-3) | 8.15E-3(1.02E-3) | 1.67E-1(2.08E-2) | 3.33E-2(4.16E-3) |
| DIMP2 | (5,10) | **2.01E+0(2.51E-1)** | 2.09E+0(2.61E-1) | 1.29E+1(1.61E+0) | 1.23E+1(1.54E+0) | 4.11E+0(5.14E-1) | 1.12E+1(1.40E+0) | 9.80E+0(1.23E+0) |
| | (10,10) | **3.03E-1(3.79E-2)** | 5.97E-1(7.46E-2) | 1.13E+1(1.41E+0) | 1.07E+1(1.34E+0) | 1.64E+0(2.05E-1) | 8.81E+0(1.10E+0) | 8.53E+0(1.07E+0) |
| | (20,10) | **1.07E-1(1.34E-2)** | 2.78E-1(3.47E-2) | 9.96E+0(1.25E+0) | 8.53E+0(1.07E+0) | 3.69E-1(4.61E-2) | 6.62E+0(8.28E-1) | 8.25E+0(1.03E+0) |
| JY2 | (5,10) | **1.31E-2(1.64E-3)** | 3.19E-2(3.99E-3) | 8.40E-2(1.05E-2) | 2.05E-1(2.57E-2) | 2.46E-2(3.07E-3) | 3.73E-1(4.66E-2) | 6.83E-2(8.53E-3) |
| | (10,10) | **7.40E-3(9.25E-4)** | 1.73E-2(2.16E-3) | 4.16E-2(5.20E-3) | 7.21E-2(9.01E-3) | 1.09E-2(1.36E-3) | 1.53E-1(1.92E-2) | 4.20E-2(5.24E-3) |
| | (20,10) | **5.59E-3(6.99E-4)** | 8.62E-3(1.08E-3) | 1.74E-2(2.17E-3) | 3.27E-2(4.09E-3) | 6.61E-3(8.26E-4) | 4.18E-2(5.22E-3) | 1.33E-2(1.66E-3) |
| JY3 | (5,10) | 4.23E-2(5.29E-3) | 1.17E-1(1.46E-2) | 5.49E-2(6.87E-3) | 5.42E-2(6.78E-3) | **3.87E-2(4.84E-3)** | 2.70E-1(3.38E-2) | 1.24E-1(1.55E-2) |
| | (10,10) | **2.87E-2(3.58E-3)** | 7.46E-2(9.33E-3) | 4.45E-2(5.56E-3) | 3.78E-2(4.72E-3) | 3.18E-2(3.98E-3) | 1.50E-1(1.87E-2) | 5.75E-2(7.19E-3) |
| | (20,10) | **2.31E-2(2.89E-3)** | 4.42E-2(5.52E-3) | 3.41E-2(4.26E-3) | 2.38E-2(2.98E-3) | 2.83E-2(3.54E-3) | 7.33E-2(9.16E-3) | 4.36E-2(5.45E-3) |
| JY6 | (5,10) | **1.04E+0(1.30E-1)** | 1.79E+0(2.23E-1) | 4.39E+0(5.48E-1) | 4.36E+0(5.45E-1) | 2.61E+0(3.26E-1) | 9.16E+0(1.14E+0) | 4.08E+0(5.10E-1) |
| | (10,10) | **2.29E-1(2.86E-2)** | 7.12E-1(8.90E-2) | 3.25E+0(4.07E-1) | 3.04E+0(3.80E-1) | 1.85E+0(2.32E-1) | 6.41E+0(8.02E-1) | 3.29E+0(4.11E-1) |
| | (20,10) | **4.69E-2(5.86E-3)** | 2.36E-1(2.95E-2) | 2.69E+0(3.36E-1) | 2.45E+0(3.07E-1) | 9.18E-1(1.15E-1) | 4.26E+0(5.33E-1) | 2.77E+0(3.46E-1) |
| JY9 | (5,10) | **3.38E-1(4.23E-2)** | 1.01E+0(1.26E-1) | 3.05E+0(3.81E-1) | 1.45E+0(1.81E-1) | 5.86E-1(7.33E-2) | 1.70E+0(2.12E-1) | 1.80E+0(2.25E-1) |
| | (10,10) | **5.32E-2(6.65E-3)** | 2.88E-1(3.60E-2) | 1.36E+0(1.70E-1) | 1.05E+0(1.31E-1) | 2.02E-1(2.53E-2) | 1.20E+0(1.50E-1) | 9.79E-1(1.22E-1) |
| | (20,10) | **3.51E-2(4.39E-3)** | 5.41E-2(6.76E-3) | 4.74E-1(5.93E-2) | 3.90E-1(4.87E-2) | 5.79E-2(7.23E-3) | 7.31E-1(9.13E-2) | 9.68E-1(1.21E-1) |

DMOEA-DVC still manages to obtain best performance on this problem thanks to the decision variable classification in change response.

In addition, from the experimental results shown in Table I, it is observed that the change frequency has a significant impact on the performance of the algorithms except DMOEA-DVC, which demonstrates the robustness of DMOEA-DVC.

Besides Table I, the average logarithmic *IGD* values of the last generation of every time step in each of the 30 independent experiments ($n_t$=10, $\tau_t$=10) on some representative problems

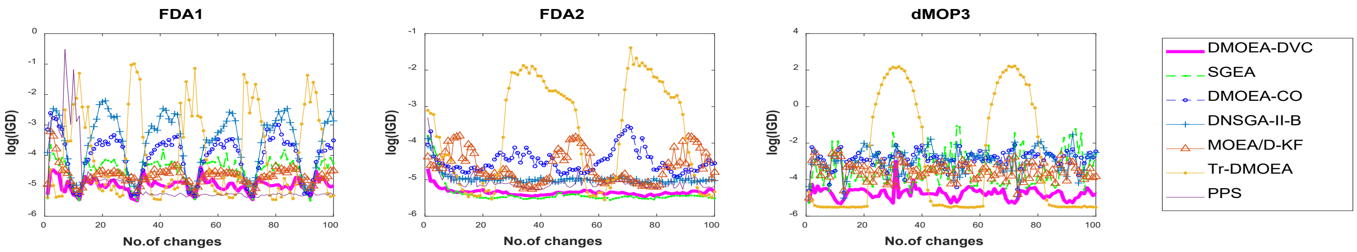
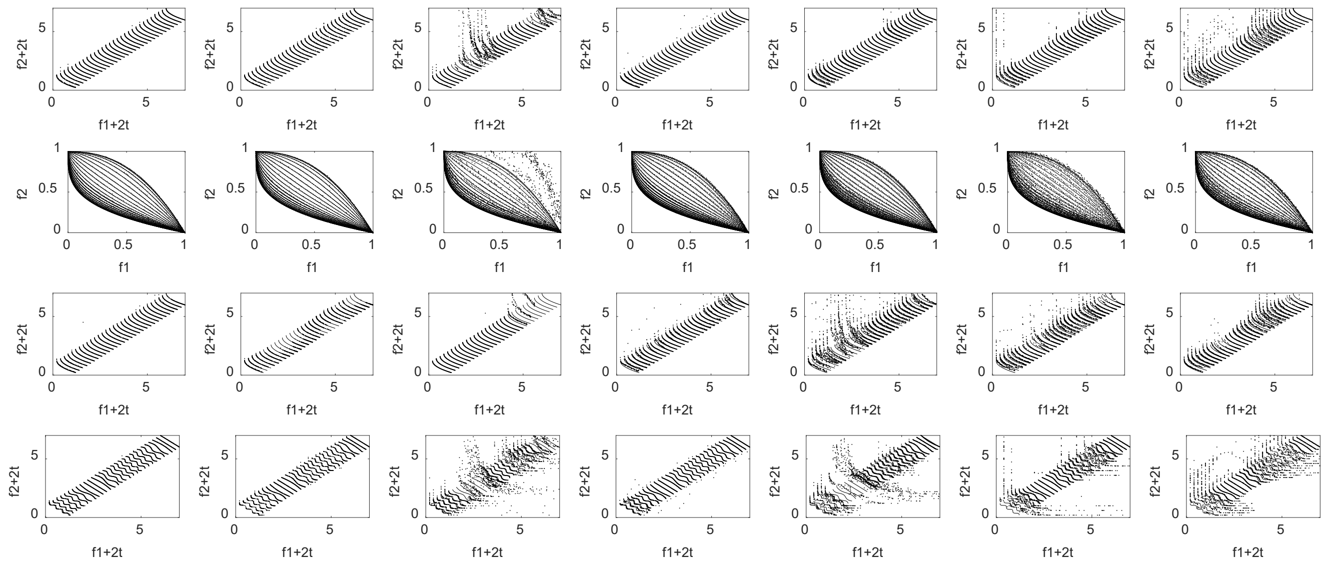



Fig. 3. Evolution curves of average log($IGD$) values for FDA1, FDA2 and dMOP3 with $\tau_t = 10$ and $n_t = 10$.




(a)DMOEA-DVC (b) SGEA (c)Tr-DMOEA (d)MOEA/D-KF (e) PPS (f) DMOEA-CO (g) DNSGA-II-B

Fig. 4. Obtained $PF(t)$ on FDA1, FDA2, dMOP3, and JY2 with $\tau_t = 10$ and $n_t = 10$. The subfigures (a), (b), (c), (d), (e), (f) and (g) show the results of DMOEA-DVC, SGEA, Tr-DMOEA, MOEA/D-KF, PPS, DMOEA-CO and DNSGA-II-B, respectively (each column per algorithm). The line and the point represent the real $PF(t)$ and $PF(t)$ identified by each algorithm, respectively.

are also plotted in Fig. 3. The curves of more classic DMOPs are provided in Fig. S1 of the Supplementary Materials. DMOEA-DVC is shown to converge quickly and maintain good diversity on most problems. Owning to the use of AR prediction model, which requires a training cycle, PPS gets significant improvement in later iterations on FDA1 and JY2. On the other hand, since the training cycle is computationally inefficient, the overall performance of PPS is not satisfactory. DMOEA-DVC requires less training time to attain better performance.

To better show the $PF(t)$ tracking capability of each algorithm, we also plot the populations obtained by each algorithm in the first 30 changes on some representative problems in Fig. 4. It is clearly observed that DMOEA-DVC has better overall tracking capability than the other algorithms on most problems.

The HV results on the classic DMOPs are provided in Tables SIX-SX of the Supplementary Materials, where DMOEA-DVC consistently shows better overall performance on FDA, dMOP, DIMP, and JY problems.

### *E. Results on New DMOPs*

The DF benchmark was proposed recently in CEC2018 Competition on Dynamic Multi-objective Optimization. From Table II, we can see that, DMOEA-DVC performs best on two-thirds of the DF problems in terms of *IGD* value. On DF2, DMOEA-DVC is worse than Tr-DMOEA, because like FDA4, the $PF(t)$ of DF2 is unchanged. Unlike the FDA, dMOP, and DIMP problems, some of the DF problems (i.e., DF3, DF4, DF7, DF8, DF9, DF10, DF11 and DF12) have nonlinear linkages between the decision variables. DMOEA-DVC obtains the best IGD values on DF4, DF8, DF9 and DF10. All decision variables of these problems are multi-optimal variables. The variable classification in static optimization of DMOEA-DVC fails on these problems, yet the classification in change response still works leading to promising performance.

Fig. 5 shows the curves of the average logarithmic IGD values on some representative DF problems. The curves of more new DMOPs are provided in Fig. S2 of the Supplementary Materials. As can be seen that, compared with the other algorithms, the IGD value of DMOEA-DVC is more stable and recovers faster in most environment changes. The $PF(t)$ tracking capability of each algorithm on DF problems is shown in Fig. 6. DMOEA-DVC is observed to have significantly better tracking capability than the other algorithms on most DF problems. The HV results on the DF problems are reported in Table SXI of the Supplementary Materials. DMOEA-DVC also shows superiority in most of the DF problems.

TABLE II
THE *IGD* COMPARISON RESULTS ON RECENT DMOPS

| Prob. | $(\tau_t,n_t)$ | DMOEA-DVC | SGEA | DMOEA-CO | DNSGA-II-B | MOEA/D-KF | Tr-DMOEA | PPS |
|---|---|---|---|---|---|---|---|---|
| DF1 | (5,10) | **1.45E-2(1.82E-3)** | 3.38E-2(4.22E-3) | 4.21E-2(5.26E-3) | 2.71E-1(3.39E-2) | 2.85E-2(3.56E-3) | 1.64E-2(2.05E-3) | 3.97E-2(4.97E-3) |
| | (10,10) | **7.74E-3(9.68E-4)** | 1.35E-2(1.69E-3) | 2.08E-2(2.60E-3) | 4.32E-2(5.39E-3) | 1.48E-2(1.85E-3) | 8.62E-3(1.08E-3) | 1.60E-2(2.01E-3) |
| | (20,10) | **5.65E-3(7.07E-4)** | 6.76E-3(8.45E-4) | 1.09E-2(1.36E-3) | 1.92E-2(2.40E-3) | 7.33E-3(9.17E-4) | 5.66E-3(7.07E-4)= | 1.26E-2(1.57E-3) |
| DF2 | (5,10) | 2.06E-2(2.57E-3) | 2.11E-1(2.63E-2) | 9.47E-2(1.18E-2) | 1.70E-1(2.13E-2) | 9.49E-2(1.19E-2) | **4.55E-3(5.68E-4)** | 9.32E-2(1.16E-2) |
| | (10,10) | 8.03E-3(1.00E-3) | 1.61E-1(2.01E-2) | 5.03E-2(6.28E-3) | 8.17E-2(1.02E-2) | 4.43E-2(5.54E-3) | **4.18E-3(5.22E-4)** | 3.84E-2(4.80E-3) |
| | (20,10) | 5.68E-3(7.10E-4) | 9.70E-2(1.21E-2) | 1.77E-2(2.22E-3) | 1.97E-2(2.46E-3) | 1.38E-2(1.72E-3) | **4.15E-3(5.19E-4)** | 3.47E-2(4.33E-3) |
| DF3 | (5,10) | 2.32E-1(2.90E-2) | 3.86E-1(4.83E-2) | 8.44E-1(1.06E-1) | 1.40E-1(1.75E-2) | **4.54E-2(5.68E-3)** | 3.95E-1(4.94E-2) | 1.90E-1(2.38E-2) |
| | (10,10) | 1.73E-1(2.16E-2) | 3.68E-1(4.60E-2) | 1.86E-1(2.33E-2) | 6.73E-2(8.42E-3) | **2.44E-2(3.04E-3)** | 3.44E-1(4.31E-2) | 2.85E-2(3.57E-3) |
| | (20,10) | 1.00E-1(1.25E-2) | 3.61E-1(4.51E-2) | 1.09E-1(1.37E-2) | 4.91E-2(6.14E-3) | **8.86E-3(1.11E-3)** | 1.32E-1(1.65E-2) | 2.06E-2(2.57E-3) |
| DF4 | (5,10) | **4.09E-2(5.11E-3)** | 7.95E-2(9.94E-3) | 1.47E+0(1.84E-1) | 5.44E-2(6.80E-3) | 5.48E-2(6.86E-3) | 2.26E+0(2.82E-1) | 2.00E-1(2.50E-2) |
| | (10,10) | **2.56E-2(3.21E-3)** | 5.81E-2(7.26E-3) | 7.34E-2(9.18E-3) | 3.02E-2(3.78E-3) | 3.74E-2(4.68E-3) | 1.50E+0(1.88E-1) | 6.74E-2(8.43E-3) |
| | (20,10) | **1.79E-2(2.24E-3)** | 3.90E-2(4.88E-3) | 3.51E-2(4.38E-3) | 2.04E-2(2.56E-3) | 2.94E-2(3.67E-3) | 9.17E-1(1.15E-1) | 2.98E-2(3.73E-3) |
| DF5 | (5,10) | **1.52E-2(1.90E-3)** | 4.96E-2(6.20E-3) | 2.88E-1(3.59E-2) | 1.44E-1(1.81E-2) | 1.93E-2(2.41E-3) | 3.47E-1(4.33E-2) | 6.46E-2(8.08E-3) |
| | (10,10) | **8.00E-3(1.00E-3)** | 2.69E-2(3.36E-3) | 1.20E-1(1.51E-2) | 7.71E-2(9.64E-3) | 9.62E-3(1.20E-3) | 1.58E-1(1.98E-2) | 3.74E-2(4.68E-3) |
| | (20,10) | **5.91E-3(7.39E-4)** | 1.65E-2(2.06E-3) | 7.87E-2(9.84E-3) | 4.12E-2(5.15E-3) | 5.98E-3(7.47E-4)= | 3.21E-2(4.01E-3) | 1.34E-2(1.67E-3) |
| DF6 | (5,10) | **5.04E-1(6.30E-2)** | 1.62E+0(2.02E-1) | 1.14E+1(1.42E+0) | 7.29E+0(9.11E-1) | 1.61E+0(2.02E-1) | 9.81E+0(1.23E+0) | 1.39E+1(1.74E+0) |
| | (10,10) | **2.51E-1(3.14E-2)** | 7.89E-1(9.86E-2) | 8.52E+0(1.06E+0) | 5.06E+0(6.33E-1) | 9.17E-1(1.15E-1) | 7.55E+0(9.44E-1) | 1.17E+1(1.46E+0) |
| | (20,10) | **1.92E-1(2.40E-2)** | 4.72E-1(5.89E-2) | 6.20E+0(7.75E-1) | 3.64E+0(4.55E-1) | 7.77E-1(9.72E-2) | 3.67E+0(4.59E-1) | 1.09E+1(1.36E+0) |
| DF7 | (5,10) | **6.26E-2(7.82E-3)** | 1.32E+0(1.65E-1) | 7.09E-2(8.86E-3) | 9.76E-2(1.22E-2) | 1.82E-1(2.27E-2) | 4.76E+0(5.95E-1) | 7.34E-2(9.18E-3) |
| | (10,10) | 4.42E-2(5.52E-3) | 1.29E+0(1.62E-1) | 2.88E-2(3.60E-3) | **2.28E-2(2.85E-3)** | 1.68E-1(2.10E-2) | 4.58E+0(5.73E-1) | 2.87E-2(3.59E-3) |
| | (20,10) | 3.81E-2(4.76E-3) | 1.06E+0(1.32E-1) | 1.95E-2(2.44E-3) | **1.71E-2(2.14E-3)** | 1.61E-1(2.02E-2) | 3.66E+0(4.58E-1) | 2.64E-2(3.30E-3) |
| DF8 | (5,10) | **1.67E-2(2.09E-3)** | 2.46E-2(3.07E-3) | 2.67E-1(3.34E-2) | 3.19E-2(3.99E-3) | 2.75E-2(3.43E-3) | 2.05E-1(2.56E-2) | 4.87E-2(6.08E-3) |
| | (10,10) | **1.58E-2(1.98E-3)** | 2.01E-2(2.51E-3) | 1.27E-1(1.59E-2) | 2.03E-2(2.54E-3) | 2.13E-2(2.67E-3) | 1.12E-1(1.40E-2) | 1.71E-2(2.13E-3) |
| | (20,10) | **1.50E-2(1.88E-3)** | 1.70E-2(2.13E-3) | 2.48E-2(3.10E-3) | 1.80E-2(2.25E-3) | 1.90E-2(2.37E-3) | 3.50E-2(4.38E-3) | 1.55E-2(1.94E-3) |
| DF9 | (5,10) | 6.36E-1(7.95E-2) | 7.62E-1(9.53E-2) | 4.17E-1(5.21E-2) | 8.81E-1(1.10E-1) | **3.64E-1(4.55E-2)** | 5.61E-1(7.01E-2) | 6.15E-1(7.69E-2) |
| | (10,10) | **2.10E-1(2.62E-2)** | 4.48E-1(5.60E-2) | 2.97E-1(3.71E-2) | 4.09E-1(5.12E-2) | 2.61E-1(3.27E-2) | 2.90E-1(3.63E-2) | 3.86E-1(4.82E-2) |
| | (20,10) | **7.59E-2(9.49E-3)** | 3.11E-1(3.89E-2) | 2.34E-1(2.93E-2) | 2.61E-1(3.27E-2) | 2.10E-1(2.62E-2) | 1.11E-1(1.38E-2) | 3.31E-1(4.14E-2) |
| DF10 | (5,10) | **1.66E-1(2.08E-2)** | 1.76E-1(2.20E-2) | 4.27E-1(5.34E-2) | 3.35E-1(4.18E-2) | 2.11E-1(2.63E-2) | 2.71E-1(3.39E-2) | 4.05E-1(5.06E-2) |
| | (10,10) | **1.63E-1(2.04E-2)** | 1.71E-1(2.14E-2) | 3.34E-1(4.17E-2) | 2.69E-1(3.36E-2) | 2.03E-1(2.54E-2) | 2.20E-1(2.74E-2) | 3.11E-1(3.88E-2) |
| | (20,10) | **1.64E-1(2.05E-2)** | 1.67E-1(2.09E-2)= | 2.96E-1(3.70E-2) | 2.79E-1(3.48E-2) | 1.97E-1(2.46E-2) | 1.86E-1(2.32E-2) | 2.84E-1(3.56E-2) |
| DF11 | (5,10) | 9.01E-2(1.13E-2) | 1.10E-1(1.37E-2) | 8.98E-2(1.12E-2)= | 8.42E-2(1.05E-2) | 7.88E-2(9.86E-3) | **7.15E-2(8.94E-3)** | 9.27E-2(1.16E-2) |
| | (10,10) | 7.18E-2(8.98E-3) | 8.65E-2(1.08E-2) | 8.03E-2(1.00E-2) | 7.78E-2(9.73E-3) | 7.31E-2(9.13E-3)= | **6.52E-2(8.15E-3)** | 8.17E-2(1.02E-2) |
| | (20,10) | **6.47E-2(8.08E-3)** | 6.96E-2(8.69E-3) | 7.62E-2(9.53E-3) | 7.44E-2(9.31E-3) | 6.99E-2(8.74E-3) | 6.61E-2(8.26E-3) | 7.90E-2(9.87E-3) |
| DF12 | (5,10) | 3.31E-1(4.13E-2) | 3.46E-1(4.32E-2) | 4.28E-1(5.34E-2) | 2.13E-1(2.66E-2) | **1.47E-1(1.84E-2)** | 6.26E-1(7.83E-2) | 6.06E-1(7.57E-2) |
| | (10,10) | 2.40E-1(3.01E-2) | 2.38E-1(2.98E-2)= | 2.95E-1(3.68E-2) | 1.67E-1(2.09E-2) | **1.08E-1(1.34E-2)** | 5.30E-1(6.62E-2) | 4.33E-1(5.41E-2) |
| | (20,10) | 1.71E-1(2.14E-2) | 1.54E-1(1.93E-2) | 2.14E-1(2.68E-2) | 1.51E-1(1.89E-2) | **8.97E-2(1.12E-2)** | 4.87E-1(6.09E-2) | 3.36E-1(4.20E-2) |
| DF13 | (5,10) | **1.61E-1(2.02E-2)** | 2.72E-1(3.40E-2) | 3.19E-1(3.98E-2) | 5.00E-1(6.25E-2) | 2.57E-1(3.22E-2) | 1.90E+0(2.37E-1) | 5.15E-1(6.44E-2) |
| | (10,10) | **1.14E-1(1.43E-2)** | 1.53E-1(1.91E-2) | 2.24E-1(2.79E-2) | 3.49E-1(4.37E-2) | 2.42E-1(3.03E-2) | 1.01E+0(1.26E-1) | 2.07E-1(2.58E-2) |
| | (20,10) | **1.01E-1(1.26E-2)** | 1.08E-1(1.35E-2) | 2.05E-1(2.56E-2) | 2.62E-1(3.27E-2) | 2.42E-1(3.02E-2) | 4.03E-1(5.04E-2) | 1.75E-1(2.19E-2) |
| DF14 | (5,10) | **5.33E-2(6.66E-3)** | 8.07E-2(1.01E-2) | 1.04E-1(1.30E-2) | 2.01E-1(2.52E-2) | 6.77E-2(8.46E-3) | 1.28E+0(1.60E-1) | 1.49E-1(1.87E-2) |
| | (10,10) | **4.23E-2(5.28E-3)** | 5.52E-2(6.89E-3) | 7.78E-2(9.72E-3) | 1.25E-1(1.56E-2) | 5.89E-2(7.36E-3) | 1.06E+0(1.33E-1) | 8.38E-2(1.05E-2) |
| | (20,10) | **3.89E-2(4.86E-3)** | 4.23E-2(5.29E-3) | 7.49E-2(9.36E-3) | 9.31E-2(1.16E-2) | 5.47E-2(6.84E-3) | 1.02E+0(1.27E-1) | 6.62E-2(8.28E-3) |

### F. Running Time Cost

The good performance of DMOEA-DVC does not come without a price. We test the running cost of all compared DMOEAs in terms of CPU time on FDA1 (two-objective), FDA4 (three-objective), dMOP3 (severe diversity loss), and DIMP1 (hard to converge). As the results shown in Table SVII of the Supplementary Materials, DNSGA-II-B consumes less time than the other approaches. DMOEA-DVC is faster than MOEA/D-KF, PPS, and Tr-DMOEA, but uses more time than SGEA and DMOEA-CO.

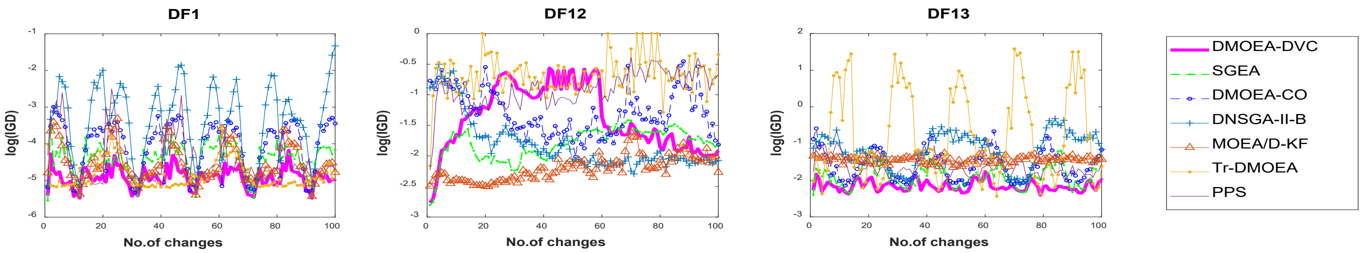
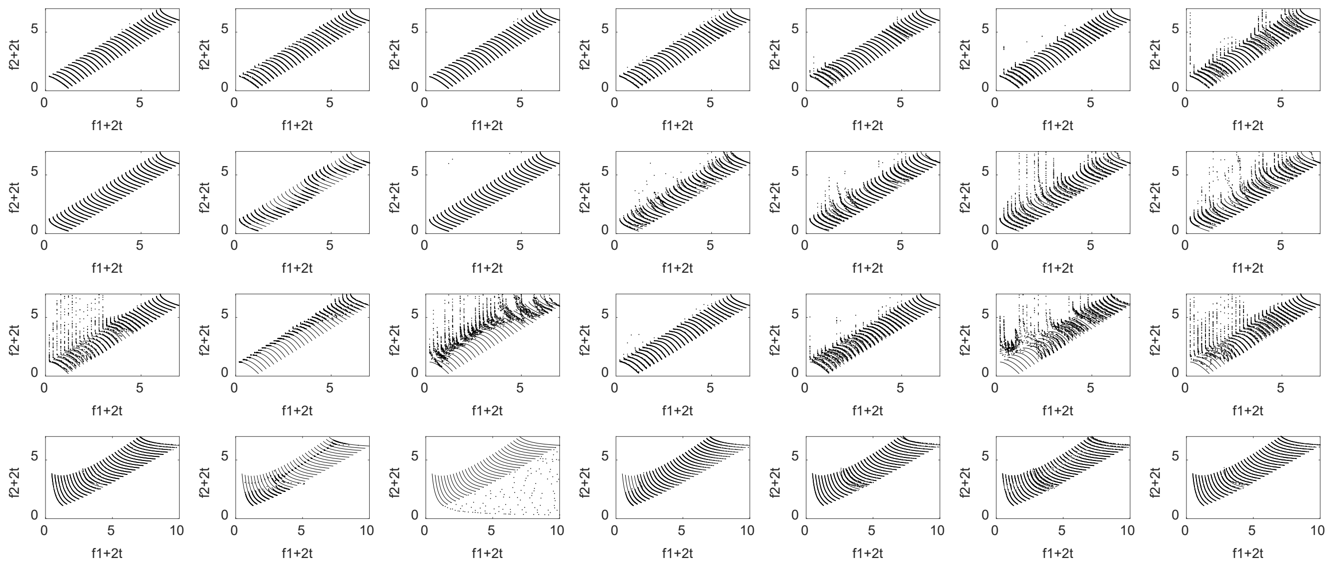



Fig. 5. Evolution curves of average log($IGD$) values on DF1, DF12 and DF13 with $\tau_t = 10$ and $n_t = 10$.




(a) DMOEA-DVC (b) SGEA (c) Tr-DMOEA (d) MOEA/D-KF (e) PPS (f) DMOEA-CO (g) DNSGA-II-B

Fig. 6. Obtained $PF(t)$ on DF1, DF2, DF3 and DF7 with $\tau_t = 10$ and $n_t = 10$. The subfigures (a), (b), (c), (d), (e), (f) and (g) show the results of DMOEA-DVC, SGEA, Tr-DMOEA, MOEA/D-KF, PPS, DMOEA-CO and DNSGA-II-B, respectively (each column per algorithm). The line and the point represent the real $PF(t)$ and $PF(t)$ identified by each algorithm, respectively.

The Table SVIII of the Supplementary Materials reports the CPU-time cost of environmental selection of DMOEA-DVC and SGEA, where ES refers to the time used by the environmental selection of the algorithms and noES refers to the time used by the algorithm excluding environmental selection. Although the environmental selection strategies used in DMOEA-DVC and SGEA are the same, the CPU-time costs of environmental selection in these two algorithms are different. The reason is that DMOEA-DVC tends to obtain more non-dominated individuals than SGEA, and it is obviously more time-consuming to select the same number of individuals from more individuals. If we ignore the time cost of environmental selection, DMOEA-DVC consumes less time than SGEA. Because the decision variable classification and other techniques introduced in DMOEA-DVC do not consume too much computational resource.

The effects of parameter $\beta$, diversity introduction and prediction, crossover operators, prediction models, and the maintenance strategy for handling tiny-changed variables are studied in the Supplementary Materials, where the configuration of DMOEA-DVC is empirically justified.

## V. Conclusions

This paper presents a DMOEA based on decision variable classification, namely DMOEA-DVC. It classifies the decision variables in both static optimization and change response stages. Different strategies are used to generate the values of different types of variables so as to achieve good balance of population diversity and convergence. DMOEA-DVC is compared with the other six state-of-the-art DMOEAs on 33 benchmark DMOPs. The experimental results show the efficiency of DMOEA-DVC.

DMOEA-DVC has shown promising performance on various benchmark problems, but it still suffers from the weakness in handling problems of slightly changed $PS(t)$ or correlated variables. To address this shortcoming, the future work could be focused on introducing advanced AR model and other promising operators in the estimation of distribution algorithm [63], particle swarm optimisation [64], [65] or other algorithms [66], [67], [68] to the DMOEA-DVC framework. Different DMOPs [69], [70] could also be considered in the future work to have a more comprehensive test of the proposed algorithm. The source code of DMOEA-DVC is publically available at http://github.com/CIA-SZU/WTC.

## References

[1] K. Deb, U. V. Rao, and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," *in Proc. EMO*, *LNCS*, vol. 4403, pp. 803-817, 2007.

[2] A. R. da Cruz, R. T. N. Cardoso, and R. H. C. Takahashi, "Multi-objective dynamic optimization of vaccination campaigns using convex quadratic approximation local search," *in Proc. EMO*, *LNCS*, vol. 6576, pp. 404-417, 2011.

[3] Z. Zhang, "Multi-objective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Appl. Soft Comput.*, vol. 8, no. 2, pp. 959-971, 2008.

[4] M. Farina, K. Deb, and P. Amato, "Dynamic multi-objective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425-442, 2004.

[5] A. Trivedi, D. Srinivasan, S. Biswas, and T. Reindl, "Hybridizing genetic algorithm with different evolution for solving the unit commitment scheduling problem," *Swarm Evol. Comput.*, vol. 23, pp.50-64, 2015.

[6] L. T. Bui, M. Zbignew, P. Eddy, and B. A. Manuel, "Adaptation in dynamic environments: A case study in mission planning," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 190-209, 2012.

[7] 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, 2002.

[8] Q. Zhang, H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712-731, 2007.

[9] K. Sindhya, K. Miettinenm, K. Deb. "A Hybrid Framework for Evolutionary Multi-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495-511, 2013.

[10] X. Qiu, J. Xu, K. C. Tan, H. A. Abbass, "Adaptive Cross-Generation Differential Evolution Operators for Multiobjective Optimization," *IEEE Trans. Evol. Comput.,* vol. 20, no. 2, pp. 232-244, 2016.

[11] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu, "On Tchebycheff Decomposition Approaches for Multiobjective Evolutionary Optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 226-244, 2018.

[12] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An Indicator Based Multi-Objective Evolutionary Algorithm with Reference Point Adaptation for Better Versatility," *IEEE Trans. Evol. Comput.*, DOI: 10.1109/TEVC.2017.2749619.

[13] L. M. Antonio, C. A. C. Coello "Coevolutionary Multi-objective Evolutionary Algorithms: A Survey of the State-of-the-Art," *IEEE Trans. Evol. Comput.*, DOI: 10.1109/TEVC.2017.2767023.

[14] R. Cheng, M. Li, K. Li, X. Yao, "Evolutionary Multiobjective Optimization Based Multimodal Optimization: Fitness Landscape Approximation and Peak Detection," *IEEE Trans. Evol. Comput.*, DOI: 10.1109/TEVC.2017.2744328.

[15] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173-195, 2000.

[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," *in Proc. IEEE Congr. Evol. Comput. (CEC)*, pp. 825-830, 2002.

[17] Q. Zhang et al., "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Elect. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-887, 2008.

[18] X. Li, S. Zhang and K. C. Wong, "Multiobjective Genome-Wide RNA-Binding Event Identification From CLIP-Seq Data," *IEEE Trans. Cybern.*, DOI: 10.1109/TCYB.2019.2960515.

[19] C. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103-127, 2009.

[20] M. Helbig and A. P. Engelbrecht, "Heterogeneous dynamic vector evaluated particle swarm optimisation for dynamic multi-objective optimisation," *in Proc. IEEE Congr. Evol. Comput. (CEC)*, pp. 3151-3159, 2014.

[21] A. P. Engelbrecht, "Heterogeneous particle swarm optimization," *in Proc. International Conference on Swarm Intelligence Springer-Verlag*, pp. 191-202, 2010.

[22] M. de Oca et al., "Heterogeneous particle swarm optimizers," *in Proc. IEEE Congr. Evol. Comput. (CEC)*, pp. 698-705, 2009.

[23] M. Greeff and A. P. Engelbrecht, "Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation," *in IEEE Congr. Evol. Comput. (CEC)*. pp. 2917-2924, 2008.

[24] M. Martínez-Peñaloza, E. Mezura-Montes, "Immune generalized differential evolution for dynamic multi-objective optimization problems," *in Proc. IEEE Congr. Evol. Comput. (CEC)*, 2015.

[25] A. Zhou, 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, 2014.

[26] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multi-objective optimization via kalman filter prediction," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2862-2873, 2015.

[27] Hatzakis, Iason, and D. Wallace, "Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach," *in Proc. Conference on Genetic and Evolutionary Computation ACM*, pp. 1201-1208, 2006.

[28] Z. Peng, J. Zheng, J. Zou, and M. Liu, "Novel prediction and memory strategies for dynamic multi-objective optimization," *Soft Comput.*, vol. 19, no. 9, pp. 2633-2653, 2014.

[29] J. Wei and Y. Wang, "Hyper rectangle search based particle swarm algorithm for dynamic constrained multi-objective optimization problems," *in Proc. IEEE Congr. Evol. Comput. (CEC)*, pp. 259-266, 2012.

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

[31] Y. Wu, Y. Jin, X. Liu, "A directed search strategy for evolutionary dynamic multi-objective optimization," *Soft Comput.* vol. 19, no. 11, pp. 3221-3235, 2015.

[32] Y. Ma, R. Liu, and R. Shang, "A hybrid dynamic multi-objective immune optimization algorithm using prediction strategy and improved differential evolution crossover operator," *in Proc. Neural Inform. Process.*, vol. 7063, pp. 435-444, 2011.

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

[34] S. B. Gee, K. C. Tan, C. Alippi, "Solving Multiobjective Optimization Problems in Unknown Dynamic Environments: An Inverse Modeling Approach," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4223-4234, 2017.

[35] M. Jiang, Z. Huang, L. Qiu, W. Huang, G. Yen, "Transfer Learning based Dynamic Multiobjective Optimization Algorithms," *IEEE Trans. Evol. Comput.*, DOI: 10.1109/TEVC.2017.2771451.

[36] R. Chen, K. Li and X. Yao, "Dynamic Multiobjectives Optimization With a Changing Number of Objectives," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 157-171, 2018.

[37] C. R. B. Azevedo and A. F. R. Araujo, "Generalized immigration schemes for dynamic evolutionary multiobjective optimization," *in Proc. IEEE Congr. Evol. Comput. (CEC)*, pp. 2033-2040, 2011.

[38] R. Azzouz, S. Bechikh, L. B. Said, "A Multiple Reference Point-based evolutionary algorithm for dynamic multi-objective optimization with undetectable changes," *in Proc. IEEE Congr. Evol. Comput. (CEC)*, pp.3168-3175, 2014.

[39] R. Shang, L. Jiao, Y. Ren, L. Li, L. Wang, "Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization," *Soft Comput.,* vol. 18, no. 4, pp. 743-756, 2014.

[40] 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.

[41] W. Koo, C. Goh, and K. C. Tan, "A predictive gradient strategy for multi-objective evolutionary algorithms in a fast changing environment," *Memetic Comput.*, vol. 2, no. 2, pp. 87-110, 2010.

[42] S. Jiang and S. Yang, "Evolutionary dynamic multi-objective optimization: benchmarks and algorithm comparisons," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 198-211, 2017.

[43] S. Jiang, S. Yang, X. Yao and K. C. Tan. "Benchmark Functions for the CEC'2018 Competition on Dynamic Multiobjective Optimization,". *Technical Report, Newcastle University, U.K.*, 2018.

[44] A. Muruganantham, Y. Zhao, S. B. Gee, X. Qiu, and K. C. Tan, "Dynamic multi-objective optimization using evolutionary algorithm with Kalman filter," *Procedia Comput. Sci.*, vol. 24, pp. 66-75, 2013.

[45] X. Zhang, Y. Tian, R. Cheng and Y. Jin, "A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-Objective Optimization." *IEEE Trans. Evol. Comput.*, vol. 22, no.1, pp. 97-112, 2018.

[46] X. Ma, F. Liu, Y. Qi, et al., "A Multiobjective Evolutionary Algorithm Based on Decision Variable Analyses for Multiobjective Optimization Problems With Large-Scale Variables." *IEEE Trans. Evol. Comput.,* vol. 20, no. 2, pp. 275-298, 2016.

[47] C. K. Goh, K. C. Tan, D. S. Liu, et al., "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design." *European Journal of Operational Research,* vol. 202, no. 1, pp. 42-54, 2010.

[48] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.,* vol. 18, no. 3, pp. 378-393, 2014.

[49] J. Sun, H. Dong, "Cooperative co-evolution with correlation identification grouping for large scale function optimization," *International Conference on Information Science and Technology (ICIST),* pp. 889–893, 2013.

[50] M. N. Omidvar, X. Li, X. Yao, "Cooperative Co-evolution with Delta Grouping for Large Scale Non-separable Function Optimization" *IEEE Congress on Evolutionary Computation,* pp. 1762–1769, 2010.

[51] Y. G. Woldesenbet, G. G. Yen, "Dynamic evolutionary algorithm with variable relocation." *IEEE Trans. Evol. Comput.,* vol. 13, no. 3, pp. 500-513, 2009.

[52] B. Xu, Z. Yong, D. Gong, et al., "Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization." *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 15, no. 6, pp. 1877-1890, 2017.

[53] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A multiobjective evolutionary algorithm based on decision variable analyses for multi-objective optimization problems with large scale variables," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275-298, 2016.

[54] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multi-objective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477-506, 2006.

[55] B. Student, "THE PROBABLE ERROR OF A MEAN," *Biometrika*, vol. 6, no. 1, pp. 1-25, 1908.

[56] D. Wang, H. Zhang, R. Liu, et al, "t-Test feature selection approach based on term frequency for text categorization," *Pattern Recognition Letters,* vol. 45, no. 1, pp. 1-10, 2014.

[57] B. Chen, W. Zeng, Y. Lin, and D. Zhang, "A new local search based multi-objective optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 50-73, 2015.

[58] C. Chen and L. Y. Tseng, "An improved version of the multiple trajectory search for real value multi-objective optimization problems," *Eng. Optim.*, vol. 46, no. 10, pp. 1430-1445, 2014.

[59] C. Rossi, M. Abderrahim, and J. C. Díaz, "Tracking moving optima using Kalman-based predictions," *Evol. Comput.*, vol. 16, no. 1, pp. 1-30, 2008.

[60] Y. Liu, D. Gong, S. Jing, et al. "A Many-Objective Evolutionary Algorithm Using A One-by-One Selection Strategy," *IEEE Trans. Cybern.,* vol. 47, no. 9, pp. 2689-2702, 2017.

[61] Z. He, G Yen, *"Many-Objective Evolutionary Algorithm: Objective Space Reduction and Diversity Improvement," IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 1-1, 2015.

[62] F. Wilcoxon, "Individual comparisons by ranking methods," *Proceedings of the Biometrics Bulletin*, vol. 1, no. 6, pp. 80-83, 1945.

[63] P. Larrañaga and J. A. Lozano, Eds., "Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation," *Kluwer Academic Publishers*, vol. 64, no. 5, pp. 1140-1148, 2001.

[64] M. Clerc, J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58-73, 2002.

[65] Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, J. Chen, C. A. C. Coello, K. Wong and J. Zhang, "Particle Swarm Optimization With a Balanceable Fitness Estimation for Many-Objective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 34-46, 2018.

[66] Q. Zhu, Q. Lin, Z. Du, Z. Liang, W. Wang, Z. Zhu, J. Chen, P. Huang and Z. Ming, "A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm," *Information Sciences*, vol. 345, pp. 177-198, 2016.

[67] K. Li, A. Fialho, S. Kwong and Q. Zhang, "Adaptive operator selection with bandits for multiobjective evolutionary algorithm based decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114-130, 2014.

[68] Z. Liang, K. Hu, Q. Zhu and Z. Zhu, "An enhanced artificial bee colony algorithm with adaptive differential operators," *Appl. Soft Comput.*, vol. 58, pp. 480-494. 2017.

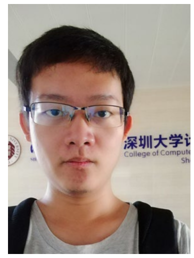
[69] M. Rong, D. Gong, W. Pedrycz and L. Wang, "A Multi-Model Prediction Method for Dynamic Multi-Objective Evolutionary Optimization," *IEEE Trans. Evol. Comput*., DOI: 10.1109/TEVC.2019.2925358.

[70] D. Gong, B. Xu, Y. Zhang, Y. Guo and S. Yang, "A Similarity-based Cooperative Co-evolutionary Algorithm for Dynamic Interval Multi-objective Optimization Problems," *IEEE Trans. Evol. Comput*., DOI: 10.1109/TEVC.2019.2912204.

**Zhengping Liang** received his B.S. degree in computer science and technology from Hunan Normal University, P,R. China, in 2001, and the Ph.D degree in computer science and technology from Wuhan University, P,R. China, in 2006. He is currently an associate professor with the college of computer science and software engineering, Shenzhen University, P,R. China.

His main research interest include computational intelligence, multi-objective optimization and big data analysis.

.




**Tiancheng Wu** received the M.S. degree from Shenzhen University, Shenzhen, China, in 2019. He is currently an algorithm engineer in 2012 laboratory of Huawei.

His current research interests include evolutionary computation, multi-objective optimization and multi-objective Neural Architecture Search.

**Xiaoliang Ma** received the B.S. degree in computing computer science and technology from Zhejiang Normal University, China in 2006, and the Ph.D. degree from the School of Computing, Xidian University, Xian, China, in 2014. He is currently an assistant professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

His current research interests include evolutionary computation, multi-objective optimization, cooperative coevolution, bioinformatics.

**Zexuan Zhu** (M'12) received the B.S. degree in computer science and technology from Fudan University, China, in 2003 and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2008. He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, China.

His research interests include computational intelligence, machine learning, and bioinformatics. Dr. Zhu is an Associate Editor of IEEE Transactions on Evolutionary Computation and IEEE Transactions on Emerging Topics in Computational Intelligence. He serves as the Editorial Board Member of Memetic Computing Journal. He is also the Chair of the IEEE CIS, Emergent Technologies Task Force on Memetic Computing.

**Shengxiang Yang** (M'00--SM'14) received the Ph.D. degree from Northeastern University, Shenyang, China in 1999. He is currently a Professor in Computational Intelligence and Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 290 publications with over 10,000 citations and an H-index of 53 according to Google Scholar. His current research interests include evolutionary computation, swarm intelligence, artificial neural networks, data mining and data stream mining, and relevant real-world applications. He serves as an Associate Editor/Editorial Board Member of a number of international journals, such as the IEEE Transactions on Evolutionary Computation, Information Sciences, Enterprise Information Systems, and Soft Computing.