A Decision Variable Classification-based Cooperative Coevolutionary Algorithm for Dynamic Multiobjective Optimization

Huipeng Xie\textsuperscript{a}, Juan Zou\textsuperscript{a,\ast}, Shengxiang Yang\textsuperscript{c,\ast}, Jinhua Zheng\textsuperscript{a,b}, Junwei Ou\textsuperscript{a}, Yaru Hu\textsuperscript{a}

\textsuperscript{a}Key Laboratory of Intelligent Computing and Information Processing (Ministry of Education), Xiangtan University, Xiangtan, Hunan, 411105, China
\textsuperscript{b}Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang, 421002, China
\textsuperscript{c}School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.

Abstract

This paper proposes a new decision variable classification-based cooperative coevolutionary algorithm, which uses the information of decision variable classification to guide the search process, for handling dynamic multiobjective problems. In particular, the decision variables are divided into two groups: convergence variables and diversity variables, and different strategies are introduced to optimize these groups. Two kinds of subpopulations are used in the proposed algorithm, i.e., the subpopulations that represent diversity variables (DS) and the subpopulations that represent convergence variables (CS). In the evolution process, the coevolution of DS and CS is carried out through the genetic operators, and subpopulations of CS are gradually merged into DS, which is optimized in the global search space, based on an indicator to avoid becoming trapped in local optimum. Once a change is detected, a prediction method and a diversity introduction approach are adopted for these two kinds of variables to get a promising population with good diversity and convergence in the new environment. The proposed algorithm is tested on 16 benchmark DMOPs.

\*Corresponding author: Shengxiang Yang and Juan Zou

Email addresses: 1264695843@qq.com (Huipeng Xie), zoujuan@xtu.edu.cn (Juan Zou), syang@dmu.ac.uk (Shengxiang Yang), junweiou@163.com (Junwei Ou), huyaru1998@gmail.com (Yaru Hu)
with state-of-the-art algorithms. Experimental results show that the proposed algorithm is very competitive for dynamic multiobjective optimization.

**Keywords:** Dynamic multiobjective optimization, Evolutionary algorithms, Decision variable classification, Cooperative coevolution.

### 1. Introduction

Many real-world multiobjective optimization problems (MOPs) possess the characteristic of being dynamic, such as scheduling [12][13][14], mineral processing [5], and wireless sensor network design [5], which are called dynamic multiobjective optimization problems (DMOPs). Due to their dynamism and multiobjective features, DMOPs pose big challenges to evolutionary algorithms (EAs). EAs need to track the time-variant Pareto optimal front (POF) or Pareto optimal set (POS) in the changing environment. The dynamic characteristics of DMOPs are reflected in dynamic objective functions [6][7][8], or dynamic constraints [9], and/or dynamically changing decision variables [10], or even a dynamically changing number of objectives [11].

Evolutionary multiobjective optimization (EMO) has drawn increasing interest and significant contributions [12][13][14][15][16] have been made to this field over the past decades, and it has been applied to many fields [17][18]. It has been established that multiobjective evolutionary algorithms (MOEAs) are powerful optimization tools to address problems with two or more conflicting goals. However, when facing DMOPs, the population that an MOEA searched may not be optimal in the new environment. Therefore, the optimization goal is not only to find the POF and POS but also to track the moving POF and/or POS. In recent years, many studies have been made to the evolutionary dynamic multiobjective optimization (EDMO) [19][20][21]. The important aspects of EDMO include change detection [22][23], construction of benchmark test problems [24][7][6][25], performance metrics [24][26][27] and algorithm design [28][29][30][31][32][33]. Among these, algorithm design for dealing with DMOPs is the main focus of this field. Due to the characteristics of DMOPs, the design of a dynamic MOEA
(DMOEA) is different from that of a static MOEA. Especially, an ideal DMOEA must possess the fast convergence ability to track the POS effectively by incorporating additional dynamic handling mechanisms and/or through the design of the evolutionary phase [1, 31, 35]. However, a fast convergence may cause a rapid loss of diversity during the optimization process, which means that it is necessary to take population diversity into account while considering fast convergence.

In recent literature, various approaches have been proposed to solve DMOPs, including prediction-based [36, 31, 34, 29, 37], memory-based [30, 38, 39], and diversity-based methods [1, 28, 40] and other approaches [41, 40, 42]. Prediction-based approaches use prediction models to predict the POS in the new environment. Memory-based approaches reuse the history information by retaining some outstanding individuals directly or maintaining some center points. Diversity-based approaches introduce some random reinitialized or mutated individuals or retain a certain proportion of distributed individuals to improve the diversity.

In addition, the static optimization part of the algorithm is also very important for the optimization of DMOPs, since after the dynamic response, the algorithm relies on the ability of static evolution to find a set of optimal solutions. Particularly, the multiple population coevolutionary mechanism [28, 43, 35] and steady-state strategy [31] are suitable for EDMO because of the high speed of convergence to quickly react to the changing environment. And cooperative coevolutionary algorithm [28, 44] is fast at convergence due to this subpopulation cooperative coevolution.

However, most of these algorithms do not take into account the different characteristics of decision variables mapped onto the objective space. As a consequence, it may be difficult for them to obtain population diversity while maintaining convergence. In this paper, a new algorithm, called decision variable classification-based cooperative coevolutionary EA (CCEA-DVC) is proposed for handling dynamic multiobjective optimization. In CCEA-DVC, the decision variables are divided into two groups: convergence variables and diversity...
variables. The purpose of the classification is to identify whether a variable affects the convergence in the optimization process. To balance diversity and convergence during the optimization process, different strategies are designed to optimize different groups of variables, and the cooperative coevolutionary process in CCEA-DVC differs from the traditional one in some ways. If a change is detected, CCEA-DVC uses an environmental selection strategy to preserve a portion of good individuals and the prediction operation is conducted on the convergence related variables, while others are unchanged. After that, a portion of individuals is randomly selected from the remaining individuals, and a diversity introduction operation is conducted on the diversity related variables. The others will enter the prediction operation. To alleviate the effect of prediction errors caused by sharp changes or something else, some randomly generated solutions will also be introduced. Generally speaking, the main contributions of this study are summarized as follows.

(1) A decision variable classification strategy is adopted, and the variables are divided into two types according to the characteristic of the variables mapping to the objective space. The classification information is used to guide the processes of optimization and dynamic response of the algorithm.

(2) A new cooperative coevolutionary method is developed to maintain good diversity while taking advantage of the rapid convergence of traditional cooperative coevolution. In this method, two different ways of cooperative coevolution are designed to optimize the subpopulations.

(3) To quickly react environmental change, a hybrid response strategy of prediction, diversity maintenance and diversity introduction is proposed to conduct on the two kinds of decision variables.

The remainder of this paper is organized as follows: Section 2 provides a brief introduction of some basic definitions and a review of the related works. The details of CCEA-DVC are presented in Section 3. The test instances, compared algorithms, performance metrics and parameter settings are introduced in
Section 4. To evaluate the performance of CCEA-DVC in addressing DMOPs, a series of comparative experiments are performed and analyzed in Section 5. A further discussion of the algorithms is offered in Section 6. Finally, Section 7 concludes this paper with discussions on future work.

2. Background

2.1. Dynamic multiobjective optimization

DMOPs can be defined in different ways, according to the nature of the dynamism they exhibit. In this paper, we mainly focus on the following type of DMOPs:

\[
\begin{align*}
\min F(x, t) &= (f_1(x, t), f_2(x, t), \ldots, f_m(x, t))^T, \\
\text{subject to } x &\in \Omega_x,
\end{align*}
\]

where \( m \) is the number of objectives; \( t \) represents the time variable; \( x = (x_1, x_2, \cdots, x_n)^T \) is the decision variable vector in which \( n \) is the number of decision variables, and \( \Omega_x \) presents the decision space. \( F(x, t) \) is the objective functions vector consisting of \( m \) objective functions that are in conflict with each other and change with \( t \). \( t \) is calculated as follows [24] [41]:

\[
t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor,
\]

where \( \tau \) is the generation number; \( n_t \) is change severity and \( \tau_t \) is change frequency.

**Definition 1. Pareto Dominance** [12]: Assume that \( p \) and \( q \) are any two individuals in the population; \( p \) is said to dominate \( q \), written as \( f(p) \prec f(q) \) if \( f_i(p) \leq f_i(q) \) \( \forall i \in 1, 2, \ldots, m \) and \( f_j(p) < f_j(q) \) \( \exists j \in 1, 2, \ldots, m \).

**Definition 2. Pareto Optimal Set (POS):** \( x \) is the decision vector; \( \Omega \) is the decision space; \( F \) is the objective function. A solution is said to be nondominated if it is not dominated by any other solutions in \( \Omega \). Thus, the POS [24] [32] is the set of all nondominated solutions and can be defined mathematically as follows:

\[
POS := \{ x \in \Omega | \nexists x^* \in \Omega, F(x^*) \prec F(x) \}.
\]
Definition 3. Pareto Optimal Front (POF): $x$ is the decision vector; $\Omega$ is the decision space; $F$ is the objective function. Thus, the POF is the set of all nondominated solutions with respect to the objective space and can be defined mathematically as follows:

$$POF := \{ y = F(x) | x \in POS \}.$$  

Due to the dynamic change of the POS and POF, Farina et al. [24] classified DMOPs into four different types.

- **Type I**: The POS changes with time but the POF is fixed.
- **Type II**: Both the POS and POF change with time.
- **Type III**: The POS remains fixed, while the POF changes with time.
- **Type IV**: Both the POS and POF remain fixed.

We mainly deal with the first three types of changes in dynamic multiobjective optimization, although the Type IV change may also occur in some cases.

2.2. Related works

In the literature, different methods have been proposed for dynamic multiobjective optimization. The existing approaches can be mainly divided according to their algorithm behaviors into two categories: convergence-based methodologies and diversity-based methodologies.

Convergence-based approaches aim to improve the convergence of the population so as to track the moving POS. Convergence-based approaches mainly include the memory strategy and prediction method. Particularly, memory strategies store optimal solutions or center point found previously and reuse them when needed [30] [38] [39]. For instance, Peng et al. [30] presented novel prediction and memory strategies (PMS) that store some non-dominated solutions into a memory pool at every time step. When an environmental change is detected, the non-dominated individuals of the memory pool in the new environment are introduced into the population. Liang et al. [38] proposed a hybrid
of memory and prediction strategies (HMPS). HMPS detects environmental changes and identifies the similarity of a change to the historical changes. If a detected change is similar to any historical changes, a memory-based technique devised to predict the new locations of the population members is applied; otherwise, a differential prediction based on the previous two consecutive population centers is adopted. The memory strategies have been shown to be very competitive for handling DMOPs. However, when historical information is not enough or environmental changes are not periodic, the performance of such strategies may not be promising.

Prediction methods try to exploit accumulated historical information to forecast the characteristics of the new environment and generate new high-quality solutions [45, 39, 29, 32], especially when environmental changes exhibit regular patterns. Hatzakis et al. [45] proposed a feed-forward prediction strategy (FPS) to predict the boundary points of the POF and then determine the location of the whole population. Zhou et al. [27] presented a population prediction strategy (PPS) to divide a POS into a centroid point and a manifold. The autoregressive (AR) model is used in FPS and PPS to predict the population. Muruganantham et al. [34] proposed a prediction model which adopts a linear discrete time Kalman filter (KF) to estimate the locations of the POS in the new environment. Jiang and Yang [31] proposed a steady-state and generational evolutionary algorithm (SGEA), in which half of the old solutions with good diversity are retained and information collected from the previous environments and the new environment is exploited to relocate the optimal solutions. In short, the prediction methods aim to enhance the algorithm’s convergence by generating a new population close to the POS of the new environment. However, in some cases, prediction error may misguide the evolution process and make the population deviate from the true POS. Furthermore, some prediction models are ineffective when lacking enough historical information.

Diversity-based approaches focus on maintaining population diversity since the changes in DMOPs may cause severe diversity loss. For instance, mutation of selected old solutions or random generation of some new ones is commonly
used to help the population escape from local optimum or improve the population diversity. Deb et al. [1] proposed a dynamic version of the non-dominated sorting genetic algorithm II (DNSGA-II) to handle DMOPs. It adopts mutated or randomly generated solutions to replace 20% of the old individuals. Goh and Tan [28] proposed a new dynamic coevolutionary algorithm (dCOEA) in which competitive and cooperative mechanisms are hybridized to solve MOPs. dCOEA uses a competitive mechanism to introduce randomly generated individuals into the subpopulation to increase diversity. Ruan et al. [32] devised an effective diversity maintenance strategy (DMS) to enhance the diversity of an MOEA. In DMS, some diverse individuals are randomly generated within the region of the next probable POS, which is defined by the maximum and minimum values in each dimension of the history center point, to prompt the diversity of the population. Appropriately enhancing population diversity is essential for algorithms to track the changing POS in the environment, while some diversity-based strategies are blind and excessive, which may negatively affect the convergence performance of the algorithm and result in evolutionary stagnation.

In addition to these approaches, some other methods have been proposed for handling DMOPs [11, 33]. Sen et al. [41] use a set of linear inverse models to guide the optimization search toward promising decision space and model objective-to-decision mapping information for fitness landscape change detection purposes. Jiang et al. [42] proposed a novel algorithm integrating traditional MOEAs and transfer learning (Tr-DMOEA), which is used to exploit the historical experience to promote the evolutionary process and to generate the initial population of the new environment. The particle swarm optimizer (PSO) can also deal with DMOPs, due to its fast convergence characteristic [46, 36].

From this discussion, it follows that the convergence-based approaches and diversity-based approaches can deal with different aspects of DMOPs, and the different components of a DMOEA make their own contributions to different aspects of solving DMOPs. In this study, a new cooperative coevolution method and a strategy that combines prediction and diversity strategies are proposed,
which aim to take advantage of each component to enhance algorithm’s ability to handle DMOPs. The proposed algorithm is described in the following section.

3. Proposed algorithm

The overall framework of the CCEA-DVC is presented in Algorithm 1 and the flowchart is presented in Figure 1. CCEA-DVC starts with an initial population $P$ and an initial archive $A$ that consists of the nondominated solutions in $P$. The decision variables are classified into two categories: convergence variables and diversity variables. Two kinds of subpopulations are used in the CCEA-DVC: the subpopulations that represent diversity variables (DS) and the subpopulations that represent convergence variables (CS). Each subpopulation of CS corresponds to a convergence variable and optimizes it inside the subpopulation, while a subpopulation, DS, is used to correspond to all diversity variables and optimize all decision variables as a whole. The distributed solutions of $P$ are used to construct the subpopulation DS, and the other solutions are used to construct CS. It should be noted that in the initial stage, computing resources (the size of subpopulation) are assigned based on the number of represented variables. In each generation cycle, when an environmental change is detected, the dynamic response (line 6 of Algorithm 1) will be activated to generate an initial population $P$, and then $P$ will be used to construct CS and DS. After that, all the subpopulations of CS coevolve together and update the representatives of each subpopulation. For simplicity, the computing resources of each variable are evenly distributed. To maximize the use of computing resources, subpopulations of CS are gradually merged into CS (line 11 of Algorithm 1) and then they participate in the optimization process of DS. Finally, an environmental selection is conducted on the combination of $A$ and $P$ and the nondominated solutions are copied into $A$ (line 13 of Algorithm 1).

Information exchange between these two kinds of subpopulations is achieved

\footnote{The truncation operation used is proposed in NSGA-II \cite{Deb2002}.}
Algorithm 1: Framework of CCEA-DVC

Input: $N$ (population size);
Output: $A$ (archive);

1. Initial parent population $P = \{x_1, x_2, ..., x_N\}$, set time period $t = 0$, set iteration generation $gen = 0$;
2. Classify decision variables by Algorithm 2;
3. Construct $DS$ (a subpopulation that represents all diversity variables) and $CS$ (subpopulations that represent convergence variables) by $P$, archive $A =$ nondominated solutions of $P$;

4. while stopping criterion not met do
   5. if change detected and not responded then
      6. ChangeResponse();
      7. $t = t + 1$;
      8. Construct $DS$ and $CS$ by $P$, archive $A =$ nondominated solutions of $P$;
   end

10. Optimize subpopulations of $CS$ in cooperative coevolution by Algorithm 3;
11. Determine whether to merge subpopulations of $CS$ into $DS$ by Algorithm 4;
12. Optimize the $DS$ by Algorithm 5;
13. $(P, A) = EnvironmentalSelection(P, A)$;
14. $gen = gen + 1$;
15. end

through the following two ways: diversity variables in $CS$ are from $DS$ and within each subpopulation of $CS$ the diversity variables are kept the same; $CS$ participates in the optimization of $DS$ through the archive $A$. The details of each component of CCEA-DVC are discussed in the following sections.
Table 1: Abbreviations and explanations commonly used in this paper

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DV</td>
<td>indices of diversity variables</td>
</tr>
<tr>
<td>CV</td>
<td>indices of convergence variables</td>
</tr>
<tr>
<td>CS</td>
<td>subpopulations that represent convergence variables</td>
</tr>
<tr>
<td>DS</td>
<td>a subpopulation that represents all diversity variables</td>
</tr>
</tbody>
</table>

3.1. Decision variable classification

In most MOEAs, the decision variables are considered as a whole during the optimizing process. Focusing on the convergence of the population may cause a loss of diversity, vice versa. So to keep a good balance of population diversity and convergence is crucial to DMOEAs. Although the decision classification is widely adopted in large-scale variables studies [47], but it is used to to group decision variables to reduce the pressure of large-scale variables optimization. To handle these issues, the decision variables are classified according to whether they are convergence related or diversity related in the objective space into two groups: convergence variables and diversity variables. A variable is called a diversity variable if and only if changing the variable while keeping others constant can only affect the non-dominating relationship between the sampling solutions. It means that the diversity variables can cause conflicts among objective functions. In contrast, if changing a variable can only result in equal, dominating or dominated relationships between the sampling solutions, the variable is called a convergence variable. Because of the operation of cooperative coevolution of DS, which is detailed in Section 3.3 variables which have both convergence and diversity properties are classified as diversity variables.

The details of the decision variables classification strategy of CCEA-DVC are presented in Algorithm 2. The count of nondominated fronts is used as a standard to classify the decision variable. For each decision variable, individual sampling is performed $n_s$ times, and the number of variables is $n$; then the total...
number of objective function evaluations of this process is $ns \times n$.

3.2. Optimization for subpopulations of CS

In the evolution process of the subpopulations of CS, the traditional cooperative coevolution is adopted to optimize convergence variables. The pseudo-code of the cooperative coevolution procedure is shown in Algorithm 3. Each sub-
Algorithm 2: Decision variable classification

Input: \(n\) (number of variables), \(ns\) (number of sampling), \(S\) (population of sampling individuals);

Output: \(DV\) (indices of diversity variables), \(CV\) (indices of convergence variables);

1. \(DV = CV = S = \emptyset\);
2. for \(i = 1\) to \(n\) do
3. Randomly select an individual \(\vec{x} = \{x_1, \ldots, x_i, \ldots, x_n\}\) from \(P\);
4. for \(j = 1\) to \(ns\) do
5. Randomly sample \(x'_i\) from its feasible region, and add \(\vec{x} = \{x_1, \ldots, x'_i, \ldots, x_n\}\) to \(S\);
6. end
7. Delete the duplicate individuals in \(S\) and calculate the objective value of the individuals in \(S\);
8. Use nondominated sorting for \(S\) to obtain the count of nondominated fronts \(c_f\);
9. if \(c_f = |S|\) then
10. \(x_i\) can be a convergence variable, \(CV \cup i\);
11. else
12. \(x_i\) can be a diversity variable, \(DV \cup i\);
13. end
14. \(S = \emptyset\);
15. end

population of CS is initialized to represent a convergence variable before the optimization process and the diversity variables are from DS. Additionally, diversity variables of all individuals in the same subpopulation are kept the same. That is to say, each subpopulation in CS represents a convergence variable; the optimization of the represented variable is inside the subpopulation, and it does not consider other variables. In the process of offspring generation, the
binary tournament selection is adapted to select mating parents, and the uniform crossover and flip mutation are chosen as the recombination and mutation operators. It should be noted that the genetic operators in this procedure are only performed on the represented variables.

**Algorithm 3: Optimization for subpopulations of CS**

**Input:** $CS$(subpopulations that represent convergence variables), $A$(archive)

**Output:** $A$(archive)

1. for $i = 1$ to $|CS|$ do /* $|CS|$ is the number of subpopulations of $CS$ */
2. Use the binary tournament selection, uniform crossover and flip mutation to create an offspring subpopulation $Q_i$ of size $|CS_i|;$
3. foreach solution $\vec{x}$ in $Q_i$, do
   4. Cooperate with other subpopulations in CS to form a complete solution $\vec{x}'$;
   5. Replace $\vec{x}$ with $\vec{x}'$;
4. end
7. $CS_i = CS_i \cup Q_i$;
8. *FastNondominateSort($CS_i$)*;
9. Update $A$ with the nondominated individuals in $CS_i$;
10. *UpdatePresent($CS_i$)*;
11. end

The cooperation process (line 4 of Algorithm 3) is to form complete individuals by concatenating the individuals in offspring subpopulation $Q_i$ with representatives from the other subpopulations in CS. For example, in Figure 2, $CS_1$, $CS_2$ and $CS_3$ represent $x_2$, $x_3$ and $x_4$ respectively. $x_1$ is a diversity variable, and $\vec{x}^{2,2}$ and $\vec{x}^{3,2}$ are the representative of $CS_2$ and $CS_3$ respectively. When $\vec{x}_1^{1,1} = (a_1, x_2, x_3, x_4)$ enters the cooperation process, $x_1$ and $x_2$ of $\vec{x}_1^{1,1}$ are unchanged; $x_3$ and $x_4$ will be obtained from $\vec{x}^{2,2}$ and $\vec{x}^{3,2}$. 
3.3. Optimization for the subpopulation DS

In order to prevent the algorithm from becoming trapped into the local optimal due to the limitations of traditional cooperative coevolution and to make full use of computing resources, CCEA-DVC uses a simple detection method to detect each subpopulation of CS to determine whether it is in the optimization.

\[^2\text{It was proposed in NSGA-II}^{12}\]

After merging \(Q_i\) into \(CS_i\), the fast nondominated sorting\[^2\] is carried out on \(CS_i\) and the nondominated individuals are used to update archive \(A\). When an individual \(\vec{x}\) enters \(A\), if \(\vec{x}\) is nondominated in \(A\), then the ones dominated by \(\vec{x}\) are deleted. Otherwise, \(\vec{x}\) is deleted. The individuals which are chosen according to nondominated fronts are retained as the parent subpopulation in the next generation. Before proceeding to the cycle of the next subpopulation, the representatives of \(CS_i\) are replaced by the nondominated individuals of \(CS_i\) to improve the speed of convergence.

Figure 2: Example of cooperation of CS. Where \(\vec{x}^{i,j}\) is the \(j\)th individual of \(CS_i\).
stagnation stage. If a subpopulation of CS is detected in the stagnation stage, it is merged into DS to make the computing resource more valuable. The details of this process are shown in Algorithm 4. Firstly, for each subpopulation of CS, the difference of the average variable value ($\delta$) between the current subpopulation and its predecessor is calculated, where $\delta$ is defined as follows:

$$
\delta = |h_{g-1} - h_g|,
$$

$$
h_g = \frac{\sum_{j=1}^{\|CS_i\|} X_j(i)}{\|CS_i\|},
$$

where $g$ is the generation number; $|CS_i|$ is the size of $CS_i$ which represents $x_{e_i}$, and $x_{e_i}$ is the $e_i$-th decision valuable; $X_j(e_i)$ is the value of $x_{e_i}$ of the $j$-th individual of $CS_i$. The $h_g$ indicates the average value of the represented variable of all the individuals of $CS_i$.

**Algorithm 4: Mergence of subpopulations of CS and DS**

**Input:** $CS$(subpopulations that represent convergence variables), $DS$(a subpopulation that represents all diversity variables)

**Output:** $A$(archive)

1. for $i = 1$ to $|CS|$ do
2. Calculate the $\delta$ of $CS_i$ using Eq. 5;
3. if $\delta < \epsilon$ then
   4. Sample the diversity variables of the individuals in $CS_i$ to improve the diversity of the subpopulation;
   5. $DS \cup CS_i$;
   6. $CS \setminus CS_i$;
4. end

If the $\delta$ of $CS_i$ is smaller than the threshold $\epsilon$, the subpopulation $CS_i$ is seen as in the stagnation stage. The diversity variables of solutions in $CS_i$ are replaced by the randomly generated value in their feasible domain, then $CS_i$ is merged into DS, and finally $CS_i$ is deleted from CS. Otherwise, it remains unchanged.
Algorithm 5: Optimization for DS

**Input:** DS (a subpopulation that represents all diversity variables), CV (indices of convergence variables), A

**Output:** DS

1. for $i = 1$ to $|DS|$ do /* $|DS|$ is the size of DS */
   2. if $rand() < 0.5$ then
      3. Randomly select an individual $a$ from $A$;
      4. Apply binary tournament selection on DS to select another two distinct individuals $b$ and $c$, which are also different from $a$;
   5. else
      6. Apply binary tournament selection on DS to select three distinct individuals $a$, $b$ and $c$ as the mating parents;
   7. end
   8. for $k = 1$ to $n$ do
      9. if $CV.IsContain(k)$ then /* whether $x^k$ is a convergence variable */
         10. $x^k = SBX(a^k, b^k)$;
      11. else
         12. $x^k = DE(a^k, b^k, c^k)$;
      13. end
   14. Update DS and A with $\vec{x}$;
   15. end

The subpopulation $DS$ evolves in a steady state manner, whenever an offspring individual is generated, it is used to update the parent population $DS$ and $A$. The cooperative coevolution between $DS$ and $CS$ is different from that of subpopulations in $CS$ (Algorithm 3). $CS$ cooperates and evolves with $DS$ by providing $DS$ with excellent convergence solutions and by archive $A$ to participate in the generation of offspring of $DS$. In Algorithm 5, $rand()$ is a randomly generated value between $(0,1)$. It should be noted that all the decision valu-
ables are optimized as a whole in the optimization process of DS to enhance the diversity and convergence of this subpopulation and help the combined subpopulation to jump out of the local optimal. Many studies and experiments show that the offspring individuals generated by SBX have a high probability of being close to their parents, but those generated by DE may deviate from their parents. This feature can be applied to different types of decision variables. In the generation process of DS, DE is used in the diversity variables and SBX is used in the convergence variables.

In this section, the population update of DS and archive A are conducted after the generation of every offspring individual $\vec{x}$. The update operation for DS actually uses $\vec{x}$ to replace the worst individual in DS according to the fitness value, after which if $\vec{x}$ is nondominated in DS, it will be used to update A. In order to make the update process more efficient, the fitness value of an individual is defined as the number of individuals that dominate it, and the update of A is just to delete the dominated individuals in it.

3.4. Change response

In order to cope with the dynamic characteristics of DMOPs, the change response mechanism must be able to track the moving optima and maintain a good level of population diversity. In the change response of most DMOEAs, the decision variables are viewed as a whole in prediction or other processes, which may mislead the population. By using the method in section 3.1, the decision variables are classified into two groups: DV and DC. Since these two groups of variables have different characteristics, different operations are conducted on them.

The overall procedure of the change response strategy (line 6 of Algorithm 1) of CCEA-DVC is presented in Algorithm 6. In change response, the population for the new environment consists of three parts. In the first part, half of the old solutions are selected from the old population by using the $k$th nearest neighbor truncation technique proposed in strength Pareto EA 2 (SPEA2). This operation is to maximize the diversity in the objective space. In this
Algorithm 6: ChangeResponse()

**Input:** $P_t, A_t, N, C_{t−1}(populationcenterofP_{t−1})$

**Output:** $P_{predict}$

1. Compute the search direction $D_t$ and the variance $\sigma_t$ using Eq. 8 and Eq. 10, respectively;
2. Select half of the old solutions from $P_t$ and use them to generate $N/2$ new individuals to form the first part named $P_{part1}$ according to Eq. 9;
3. Randomly select $N_{p2}$ old solutions from the remaining solutions in $P_t$, and use them to generate new individuals to form the second part named $P_{part2}$ according to Eq. 9 and Eq. 11;
4. Randomly initialize $N − N/2 − N_{p2}$ individuals to form the third part named $P_{part3}$;
5. $P_{predict} = P_{part1} \cup P_{part2} \cup P_{part3}$;

part, the diversity variables remain unchanged, and the others participate in the prediction process to enhance the convergence of the population.

Different from the complex prediction model used in MOEA/D-KF [34] and PPS [29], this paper mainly adopts the moving direction of the previous two consecutive population centers to predict the new location of the POS in the new environment. In order to avoid the problem of poor individuals misleading the prediction of the population, the center of nondominated solutions, which are the elitist individuals of the current population, is seen as the position of the population center. Let $A_t$ be the nondominated solutions at time $t$, and then the position of the population center can be expressed by the following formula:

$$C_t = \frac{1}{|A_t|} \sum_{x_t \in A_t} x_t^i,$$  \hspace{1cm} (7)

where $|A_t|$ is the size of nondominated solutions; $x_t^i$ is the $i$th decision variable of the individual at time $t$; $i = 1, 2, ..., n$, and $n$ is the number of decision variables. The population center referred to as $C_t = \{C_t^1, C_t^2, ..., C_t^n\}$ is a vector with $n$
dimensions. Thus, the moving direction of center points termed as \( D_t \) at time \( t \) can be defined as:

\[
D_t = C_t - C_{t-1}.
\]  

For each individual at time \( t \) which enters the prediction operation, its new location in the decision space is generated as follows:

\[
\begin{cases}
  x^i_{t+1} = x^i_t + D^i_t + N(0, \sigma_t), & \text{for all } i \in CV,
\end{cases}
\]

where \( CV \) contains the indices of convergence variables, and \( N(0, \sigma_t) \) is a Gaussian noise to increase the probability of the predicted solution to locate in the POS in the new environment. \( \sigma_t \) is defined by:

\[
\sigma_t = \left( \sum_{i=1}^{n} |D^i_t|^2 \right)^{\frac{1}{2}}.
\]  

In the second part, \( N_{p2} \) old solutions are randomly selected from the remaining solutions in the old population. In order to make full use of the classification information of decision variables and improve the diversity of the population, apart from the prediction operation, a diversity introduction strategy is proposed in the second part. First, the minus and maximum value of each variables in \( DV \) are recorded in these two vectors: \( \overrightarrow{MIN} = \{..., x^\text{min}_i, x^\text{max}_i, ...\} \) and \( \overrightarrow{MAX} = \{..., x^\text{max}_i, x^\text{max}_i, ...\} \). The diversity introduction strategy can be expressed by the following formula:

\[
\begin{cases}
  x^i_{t+1} = \text{random}(x^\text{LOW}_i, x^\text{min}_i), & \text{if } \text{rand}() < 0.5, \\
  x^i_{t+1} = \text{random}(x^\text{UP}_i, x^\text{max}_i), & \text{if } \text{rand}() \geq 0.5,
\end{cases}
\]

for all \( i \in DV \),

where \( \text{random}(a, b) \) is a function which returns a random value between \( a \) and \( b \); \( x^\text{LOW}_i \) and \( x^\text{UP}_i \) is the ith decision variable’s lower bound and upper bound, respectively; \( \text{rand}() \) is a randomly generated value between \([0,1)\), and \( DV \) is a set of indices of diversity variables.

In the third part, \( N - N/2 - N_{p2} \) individuals are randomly reinitialized to enhance population diversity and mitigate the effect of prediction errors.
It should be noted that when the first environmental change occurs, the computation of $D_t$ is not feasible. In such a situation, the polynomial mutation is applied to reinitialize the population.

4. Experimental design

4.1. Test instances

The proposed algorithm is tested on sixteen problems, including five FDA problems [24], three dMOP problems [28] and nine JY problems (JY1-JY9) [7]. The FDA benchmark suite is commonly used in the performance evaluation of DMOEAs. The dMOP benchmark problems are an extension of the FDA benchmark suite. The JY test suite is a recently proposed benchmark, which is able to tune a number of challenging characteristics, including mixed POF (convexity concavity), nonmonotonic and time-varying variable linkages, mixed types of changes, and randomness in type change. The time instance $t$ involved in these problems is defined as $t = (1/n_t)\lfloor(\tau/\tau_t)\rfloor$ (where $n_t$, $\tau_t$ and $\tau$ represent the severity of change, the frequency of change, and the iteration counter, respectively).

4.2. Performance metrics

In our experimental studies, the following performance metrics are adopted to assess the performance, including convergence, distribution, and diversity of the algorithms.

(1) Generational Distance (GD): The GD [30, 10, 12, 13] is a widely used metric which measures the convergence of the obtained solutions by an algorithm. Let $POF_t$ be a set of uniformly distributed points in the true $POF$ at time $t$, and $P_t$ be an approximation of the $POF$. The GD is defined as:

$$GD = \frac{\sum_{v \in P_t} d(POF_t, v)}{|P_t|},$$

(12)

where $d(POF_t, v) = \min_{v \in POF_t}\sqrt{\sum_{j=1}^{m}(f_j^v - f_j^w)^2}$ represents the Euclidean distance between individual $v$ in $P_t$ and its nearest individual in $POF_t$. The smaller the GD value is, the better the convergence the algorithm has.
(2) Inverted Generational Distance (IGD): The IGD \[29, 32, 37\] can measure both the convergence and diversity (including the range and the uniformity of distribution) of the obtained solution set by an algorithm. IGD is defined as:

\[
IGD = \frac{\sum_{v \in P_t} d(v, P_t)}{|P_t|},
\]

where \(d(v, P_t) = \min_{u \in P_t} \sqrt{\sum_{j=1}^{m} (f^u_j - f^v_j)^2}\) represents the Euclidean distance between individual \(v\) in POF and its nearest individual in \(P_t\).

(3) Hypervolume Difference (HVD): The HVD \[31\][29][48][49] measures the gap between the hypervolume of the obtained POF and that of the true POF:

\[
HVD = HV(POF_t) - HV(P_t).
\]

4.3. Compared algorithms and parameter settings

Four popular DMOEAs are used for comparison in our experiment, including FPS \[45\], PPS \[29\], SGEA \[31\], and DNSGA-II \[1\]. FPS uses a feed-forward prediction strategy to predict the boundary points of the POF and then locate the location of the whole population when the next environmental change occurs. The forecasting model it uses is the autoregressive (AR) model. In PPS, the optimal solution set is divided into two parts: the population center and manifold. It adopts a univariate autoregression (AR) model to predict the next population center and the initial population for the new environment is created from the predicted center and predicted manifold. DNSGA-II is a dynamic version of the popular NSGA-II \[12\] which replaces some population members with mutated solutions if a change occurs. SGEA uses the fast and steady tracking ability of the steady-state mechanism and the good diversity preservation of generational algorithms to solve DMOPs.

As for the implementations of the algorithms, PPS and SGEA are performed with the source code provided by the author. FPS and DNSGA-II are implemented with C++ based on the algorithm description of the original papers. Since all the algorithms of the experiment are basically based on C or C++,
the programming language has little influence on the experimental results. All experiments are performed with Microsoft Visual Studio 2016 on a laptop with an Intel(R) Core(TM) i5-3230M CPU and 8 GB RAM.

The parameters of the compared DMOEAs in the experiment are configured following the original references. However, some parameters are revised to make the comparison more fair, and the key parameters in these algorithms are set as follows.

(1) The population size ($N$) for all the test problems is set to 100 and the dimension of decision variables ($n$) is set to 10. The archive size is the same as the population.

(2) For CCEA-DVC, the number of sampling ($ns$) in variable classification (Algorithm 2) is set to 10. The threshold ($\epsilon$) of Algorithm 4 is set to the 5% of the difference between the upper and lower bounds of the $i$th decision variable. The size ($N_{p2}$) of $P_{part2}$ in Algorithm 6 is set to 35% of the population size.

(3) For FPS and PPS, the parameters used here are the same as the original reference. RM-MEDA [29] is chosen as the MOEA optimizer of FPS and PPS.

(4) The SBX control parameters $p_c$ and $\eta_c$ are set to 1.0 and 20, respectively. The DE control parameters $\text{CR}$ and $F$ are set to 1.0 and 0.5, respectively. The mutation probability $p_m = 1/n$, and its distribution $\eta_m = 20$.

(5) For all the algorithms, 5% of the population are randomly selected as the detectors, and they are reevaluated to detect the environmental changes.

(6) In the experiment, each algorithm is run 10 times independently on each test problem. The total number of generations for each run is set to 50$\tau_t$, which ensures 50 environmental changes in each run. It should be noted that SGEA did not run 50 generations before the first environmental change, in order to keep the experiment fair.
4.4. Computational complexity

In this section, the computational complexity of the proposed algorithm is analyzed. Let $M$ be the number of objectives, $n$ be the decision variable number and $N$ be the population number. The computing resources consumption of CCEA-DVC is mainly in terms of decision variable classification and static algorithm optimization. The total number of objective function evaluations of decision variable classification is $O(ns \cdot n)$, which has been analyzed in 3.1. In optimization of CS, the complexity of fast nondominate sorting is $O(MN^2)$. The complexity of population update procedure of DS is $O(N^2)$. The offspring generation of these two part take $O(MN)$ computations and the environmental selection part spends $O(MN^2)$ computations \[13\] on crowding distance calculation and sorting. Therefore, the overall computational complexity of CCEA-DVC in each generation is $O(MN^2)$.

5. Experimental results and analysis

To compare the effect of change frequency on the compared algorithms in dynamic environments, the severity of change ($n_t$) was fixed to 10, and the frequency of change ($\tau_t$) was set to 10, 15 and 20, respectively. The obtained average GD, IGD and HVD results over a series of time windows and their standard deviation values are presented in Tables 2–4, respectively. The best values obtained by the five algorithms are highlighted in bold face, and the Wilcoxon rank-sum test \[31\] is carried out to indicate significance between different results at the 0.05 significance level.

5.1. Results on FDA and dMOP problems

This section presents the results and analysis of the algorithms on FDA and dMOP problems. It can be observed from Table 2 that CCEA-DVC had the best results on the majority of the tested FDA and dMOP problems whose decision variables are linearly related. For most problems with different change frequencies, CCEA-DVC had better convergence than the four other algorithms.
CCEA-DVC surpassed SGEA on all FDA and dMOP problems except FDA4 with $\tau_t = 15$ and 20. FDA4 is different from the others as it is a three-objective problem which poses difficulty for algorithms in population convergence. The decision classification of CCEA-DVC is based on the characteristic of decision variables mapping to the objective space, so in the three-dimensional space, incorrect classification may occur.

For all the tested problems, PPS and FPS failed to show encouraging performance on the GD metrics as the AR prediction model adopted in them needs to be trained for a period. As the number of evolutionary generations increased at different change frequencies, the performance of DNSGA-II significantly improved, but it needed more time to converge to the POF.

As shown in Table 3, CCEA-DVC performed the best on most of the test instances except FDA4, where SGEA was the bestperformer, in terms of IGD value. Similarly, the classification error in the high-dimensional objective space is the main reason for the unsatisfactory performance of CCEA-DVC on FDA4 with $\tau_t = 15$ and 20. And the uncompetitive convergence of obtained approximations also affected the IGD value. For FDA1, FDA3 and dMOP1, CCEA-DVC achieved faster convergence through the new cooperation coevolution method, and it also obtained uniformly distributed approximations that benefit from the special operation on the diversity variables, which indicates that CCEA-DVC is very competitive in solving two-objectives problems.

The HVD values are slightly different from the IGD values on FDA and dMOP displayed in Tables 3 and 4. Specifically, for FDA2, SGEA is in first place and CCEA-DVC is in second in terms of HVD, while in terms of IGD, the situation is reversed. This is caused by the different calculation of IGD and HVD, and the difference of the IGD value is very little. For FDA3, CCEA-DVC is slightly inferior to FPS indicating that the environmental response of FPS is also competitive in handling problems with changing density distribution.

Apart from the tabular presentation, the obtained POFs of FDA1, FDA3, dMOP2 and dMOP3 over 21 or 41 time windows are shown in Figure 3.
Table 2: Mean and SD of GD indicator obtained by five algorithms.

<table>
<thead>
<tr>
<th>Prob. (n₁, n₂)</th>
<th>DNSGA-II</th>
<th>PPS</th>
<th>FPS</th>
<th>SGEA</th>
<th>CCEA-DVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10,10)</td>
<td>5.845e-2 (3.993e-2)</td>
<td>1.158e-1 (7.927e-2)</td>
<td>6.591e-2 (6.332e-3)</td>
<td>9.072e-2 (3.132e-2)</td>
<td>¿3.02e-2 (5.50e-2)</td>
</tr>
<tr>
<td>(15,10)</td>
<td>1.986e-1 (7.775e-6)</td>
<td>7.560e-2 (6.89e-6)</td>
<td>3.145e-2 (1.128e-5)</td>
<td>6.155e-2 (1.87e-5)</td>
<td>¿4.52e-2 (4.219e-5)</td>
</tr>
<tr>
<td>(20,10)</td>
<td>1.035e-2 (3.36e-4)</td>
<td>5.100e-2 (2.61e-4)</td>
<td>1.902e-2 (2.61e-4)</td>
<td>4.672e-2 (3.97e-5)</td>
<td>¿3.52e-2 (1.471e-4)</td>
</tr>
<tr>
<td>(10,15)</td>
<td>4.296e-2 (5.345e-4)</td>
<td>6.062e-2 (2.51e-3)</td>
<td>6.972e-2 (2.11e-3)</td>
<td>4.730e-2 (2.59e-4)</td>
<td>¿3.66e-2 (2.019e-3)</td>
</tr>
<tr>
<td>(20,15)</td>
<td>1.178e-1 (2.32e-2)</td>
<td>6.630e-2 (2.64e-3)</td>
<td>7.372e-2 (2.70e-3)</td>
<td>4.360e-2 (2.12e-4)</td>
<td>¿4.12e-2 (2.161e-4)</td>
</tr>
<tr>
<td>(10,10)</td>
<td>9.096e-2 (4.03e-3)</td>
<td>1.777e-1 (9.60e-2)</td>
<td>1.015e-1 (1.37e-2)</td>
<td>4.106e-2 (3.02e-3)</td>
<td>¿1.29e-2 (7.79e-2)</td>
</tr>
<tr>
<td>(15,10)</td>
<td>6.252e-2 (1.85e-3)</td>
<td>1.060e-1 (6.60e-3)</td>
<td>7.703e-2 (2.08e-3)</td>
<td>3.811e-2 (2.87e-3)</td>
<td>¿1.23e-2 (7.98e-2)</td>
</tr>
</tbody>
</table>

† and ‡ indicate CCEA-DVC performed significantly better than and equivalently to the corresponding algorithm respectively.
<table>
<thead>
<tr>
<th>Prob. ((n_1, n_2))</th>
<th>DNSGA-II</th>
<th>PPS</th>
<th>FPS</th>
<th>SGEA</th>
<th>CCEA-DVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD1 (10, 10)</td>
<td>5.306e-2</td>
<td>2.961e-2</td>
<td>4.132e-2</td>
<td>5.806e-3</td>
<td>7.642e-3</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>2.124e-1</td>
<td>5.962e-2</td>
<td>7.830e-2</td>
<td>3.318e-3</td>
<td>6.131e-3</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>1.698e-2</td>
<td>2.579e-2</td>
<td>4.257e-2</td>
<td>3.175e-3</td>
<td>5.501e-3</td>
</tr>
<tr>
<td>FD2 (15, 10)</td>
<td>3.676e-1</td>
<td>3.172e-1</td>
<td>3.327e-1</td>
<td>5.886e-2</td>
<td>2.843e-2</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>3.561e-1</td>
<td>3.258e-1</td>
<td>3.235e-1</td>
<td>2.860e-2</td>
<td>2.831e-2</td>
</tr>
<tr>
<td>FD3 (10, 10)</td>
<td>4.591e-2</td>
<td>3.333e-2</td>
<td>2.976e-2</td>
<td>1.832e-2</td>
<td>4.976e-2</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>2.709e-2</td>
<td>4.946e-2</td>
<td>4.139e-2</td>
<td>1.576e-2</td>
<td>3.917e-2</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>2.114e-2</td>
<td>3.432e-2</td>
<td>2.975e-2</td>
<td>1.243e-2</td>
<td>3.550e-2</td>
</tr>
<tr>
<td>FD4 (10, 10)</td>
<td>1.624e-1</td>
<td>1.106e-1</td>
<td>1.104e-1</td>
<td>1.080e-2</td>
<td>9.144e-2</td>
</tr>
<tr>
<td>dMOP1 (10, 10)</td>
<td>9.087e-2</td>
<td>1.044e-1</td>
<td>5.637e-2</td>
<td>1.432e-2</td>
<td>9.395e-2</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>3.582e-2</td>
<td>4.110e-2</td>
<td>1.578e-2</td>
<td>3.725e-2</td>
<td>5.690e-2</td>
</tr>
<tr>
<td>dMOP2 (10, 10)</td>
<td>5.535e-2</td>
<td>6.077e-2</td>
<td>4.496e-2</td>
<td>1.545e-2</td>
<td>7.560e-2</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>5.638e-2</td>
<td>2.906e-2</td>
<td>2.155e-2</td>
<td>8.745e-3</td>
<td>6.015e-3</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>2.685e-2</td>
<td>3.045e-2</td>
<td>1.375e-2</td>
<td>6.894e-3</td>
<td>5.278e-3</td>
</tr>
<tr>
<td>dMOP3 (10, 10)</td>
<td>5.975e-2</td>
<td>7.297e-2</td>
<td>2.548e-2</td>
<td>1.566e-2</td>
<td>6.885e-3</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>6.170e-2</td>
<td>5.843e-2</td>
<td>2.548e-2</td>
<td>1.186e-2</td>
<td>5.885e-3</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>5.394e-2</td>
<td>7.944e-2</td>
<td>5.230e-2</td>
<td>2.448e-2</td>
<td>6.024e-3</td>
</tr>
<tr>
<td>dMOP4 (10, 10)</td>
<td>3.173e-1</td>
<td>5.712e-2</td>
<td>2.735e-2</td>
<td>1.973e-2</td>
<td>5.172e-2</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>5.628e-2</td>
<td>5.958e-2</td>
<td>5.498e-2</td>
<td>5.248e-2</td>
<td>5.692e-2</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>3.529e-1</td>
<td>7.664e-2</td>
<td>5.230e-2</td>
<td>5.110e-2</td>
<td>5.060e-2</td>
</tr>
<tr>
<td>dMOP5 (10, 10)</td>
<td>3.164e-1</td>
<td>2.526e-1</td>
<td>2.548e-2</td>
<td>2.548e-2</td>
<td>6.885e-3</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>1.686e-1</td>
<td>2.413e-2</td>
<td>1.686e-2</td>
<td>2.413e-2</td>
<td>4.951e-2</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>6.261e-2</td>
<td>5.628e-2</td>
<td>5.628e-2</td>
<td>5.628e-2</td>
<td>4.951e-2</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>5.960e-2</td>
<td>5.960e-2</td>
<td>5.960e-2</td>
<td>5.960e-2</td>
<td>4.951e-2</td>
</tr>
<tr>
<td>dMOP7 (10, 10)</td>
<td>3.552e-1</td>
<td>7.122e-2</td>
<td>1.209e-2</td>
<td>1.209e-2</td>
<td>4.951e-2</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>1.251e+1</td>
<td>1.251e+1</td>
<td>1.251e+1</td>
<td>1.251e+1</td>
<td>4.951e-2</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>5.337e-2</td>
<td>5.337e-2</td>
<td>5.337e-2</td>
<td>5.337e-2</td>
<td>4.951e-2</td>
</tr>
<tr>
<td>dMOP8 (10, 10)</td>
<td>2.079e-2</td>
<td>3.190e-2</td>
<td>2.962e-2</td>
<td>3.190e-2</td>
<td>2.962e-2</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>1.630e-2</td>
<td>1.259e-2</td>
<td>1.259e-2</td>
<td>1.259e-2</td>
<td>2.962e-2</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>5.960e-2</td>
<td>5.960e-2</td>
<td>5.960e-2</td>
<td>5.960e-2</td>
<td>2.962e-2</td>
</tr>
<tr>
<td>dMOP9 (10, 10)</td>
<td>3.818e-2</td>
<td>3.818e-2</td>
<td>3.818e-2</td>
<td>3.818e-2</td>
<td>2.962e-2</td>
</tr>
<tr>
<td>(15, 10)</td>
<td>2.486e-2</td>
<td>2.486e-2</td>
<td>2.486e-2</td>
<td>2.486e-2</td>
<td>2.962e-2</td>
</tr>
<tr>
<td>(20, 10)</td>
<td>2.391e-2</td>
<td>2.391e-2</td>
<td>2.391e-2</td>
<td>2.391e-2</td>
<td>2.962e-2</td>
</tr>
</tbody>
</table>

Table 3: Mean and SD of IGD indicator obtained by five algorithms.
Figure 3: Obtained POFs for FDA1, FDA3, dMOP2 and dMOP3 problems with $n_t = 10$ and
$\tau_t = 10$
is obvious that CCEA-DVC is very capable of tracking the moving optimum in the changing environment. Particularly, if there is a diversity loss (e.g., on FDA3 and dMOP3) in the dynamic environment, CCEA-DVC can obtain the POF with good convergence and diversity. It should be pointed out that CCEA-DVC conducts diversity introduction on diversity variables in dynamic reaction, so it can perform better in such problems.

5.2. Results on JY problems

As indicated in Table 2, CCEA-DVC has the best solution convergence on four problems—JY1, JY5, JY8 and JY9—but seems to perform worse than SGEA on the remaining five test cases, although the difference is very little. Next, let’s analyze the specific reasons. In SGEA, the dynamic response and optimization process are in a steady-state manner, so SGEA possesses fast convergence ability. As for JY2, CCEA-DVC almost performs as well as SGEA and shows significantly better performance than the other three algorithms in the GD values. JY3 is a problem in which there are time-varying non-monotonic dependencies between any two decision variables, and as time increases, the POF becomes increasingly complicated. CCEA-DVC optimizes each convergence variable in a subpopulation, so it often performs poorly when facing a problem with non-monotonic dependencies. For JY4, whose POF is discontinuous and whose disconnected segments change with time, CCEA-DVC performed poorly. JY6 and JY7 are both multimodal problems, the number of local optima of JY6 changes over time but JY7’s remain fixed. In the early stages of the CCEA-DVC evolution, it may fall into a local optimal state, and even if it can be detected, the overall performance will be affected. Even so, CCEA-DVC almost performs as well as SGEA, which indicates that CCEA-DVC has a certain potential in handling multimodal problems.

The IGD metric mainly depends on the uniformity, distribution, and coverage of an approximation to the true POF. Together with GD, we can analyze the performance of the algorithm on test problems comprehensively. It can be seen
<table>
<thead>
<tr>
<th>Prob</th>
<th>(n, m)</th>
<th>DSNBGA-II</th>
<th>FPS</th>
<th>FPS</th>
<th>SGEA</th>
<th>CCEA-DVC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDA1</td>
<td>(10, 10)</td>
<td>1.060e-1</td>
<td>4.089e-2</td>
<td>8.650e-2</td>
<td>4.030e-1</td>
<td>1.935e-3</td>
</tr>
<tr>
<td>FDA2</td>
<td>(10, 10)</td>
<td>1.580e-1</td>
<td>4.620e-2</td>
<td>7.150e-2</td>
<td>5.380e-2</td>
<td>2.160e-1</td>
</tr>
<tr>
<td></td>
<td>(15, 15)</td>
<td>1.660e-1</td>
<td>5.090e-2</td>
<td>7.160e-2</td>
<td>5.390e-2</td>
<td>2.170e-1</td>
</tr>
<tr>
<td></td>
<td>(20, 20)</td>
<td>1.740e-1</td>
<td>5.550e-2</td>
<td>7.170e-2</td>
<td>5.400e-2</td>
<td>2.180e-1</td>
</tr>
<tr>
<td></td>
<td>(15, 15)</td>
<td>1.460e-1</td>
<td>5.090e-2</td>
<td>6.660e-2</td>
<td>4.630e-2</td>
<td>2.170e-1</td>
</tr>
<tr>
<td>FDA4</td>
<td>(10, 10)</td>
<td>1.580e-1</td>
<td>4.620e-2</td>
<td>7.150e-2</td>
<td>5.380e-2</td>
<td>2.160e-1</td>
</tr>
<tr>
<td></td>
<td>(15, 15)</td>
<td>1.660e-1</td>
<td>5.090e-2</td>
<td>7.160e-2</td>
<td>5.390e-2</td>
<td>2.170e-1</td>
</tr>
<tr>
<td></td>
<td>(20, 20)</td>
<td>1.740e-1</td>
<td>5.550e-2</td>
<td>7.170e-2</td>
<td>5.400e-2</td>
<td>2.180e-1</td>
</tr>
<tr>
<td>dMOH1</td>
<td>(10, 10)</td>
<td>1.580e-1</td>
<td>4.620e-2</td>
<td>7.150e-2</td>
<td>5.380e-2</td>
<td>2.160e-1</td>
</tr>
<tr>
<td></td>
<td>(15, 15)</td>
<td>1.660e-1</td>
<td>5.090e-2</td>
<td>7.160e-2</td>
<td>5.390e-2</td>
<td>2.170e-1</td>
</tr>
<tr>
<td></td>
<td>(20, 20)</td>
<td>1.740e-1</td>
<td>5.550e-2</td>
<td>7.170e-2</td>
<td>5.400e-2</td>
<td>2.180e-1</td>
</tr>
<tr>
<td>dMOH2</td>
<td>(10, 10)</td>
<td>1.580e-1</td>
<td>4.620e-2</td>
<td>7.150e-2</td>
<td>5.380e-2</td>
<td>2.160e-1</td>
</tr>
<tr>
<td></td>
<td>(15, 15)</td>
<td>1.660e-1</td>
<td>5.090e-2</td>
<td>7.160e-2</td>
<td>5.390e-2</td>
<td>2.170e-1</td>
</tr>
<tr>
<td></td>
<td>(20, 20)</td>
<td>1.740e-1</td>
<td>5.550e-2</td>
<td>7.170e-2</td>
<td>5.400e-2</td>
<td>2.180e-1</td>
</tr>
</tbody>
</table>

† and ‡ indicate CCEA-DVC performed significantly better than and equivalently to the corresponding algorithm respectively.
Figure 4: Obtained POFs for JY2, JY5, JY6 and JY7 problems with $n_t = 10$ and $r_t = 10$
from Table 3 that CCEA-DVC performed the best on most of the test instances except JY3 and JY4, where DNSGA-II and SGEA are in first place, respectively. Although CCEA-DVC performed poorly on JY4 in terms of the GD metric, it improves a lot in terms of IGD. It means that CCEA-DVC performs well in uniformity and distribution of the population. As for JY2, JY6, and JY7, SGEA wins on solution convergence, but it loses to CCEA-DVC on IGD and HVD (4), which measure both the convergence and diversity of found solutions. The HVD values in Table 4 are basically consistent with the IGD ones in Table 3. CCEA-DVC shows a little worse performance when with SGEA on JY6 with \( \tau_t = 15 \) and 20. In general, CCEA-DVC is more promising than the other algorithms in dealing with different dynamic types. It can be concluded from this analysis that performing different operations on different kinds of variables is effective in maintaining both population diversity and convergence.

In Figure 4, the obtained POFs of JY2, JY5, JY6 and JY7 over 21 time windows are plotted. It can be seen from Figure 4 that CCEA-DVC performed better than SGEA in JY6 and JY7 in population convergence and diversity, especially in JY7, although CCEA-DVC is a little worse than SGEA in these two problems in terms of GD value. The reason may be that in CCEA-DVC, the different operations conducted on different kinds of variables can assist the optimization of population convergence and population diversity in a balance.

5.3. Comparing evolutionary processes of different algorithms

In addition to providing the experimental data table, this article also displays the evolution curves of the IGD value of each algorithm on various problems with time in a superimposed manner to better analyze the specific performance of the algorithm at different times, as shown in Figure 5, Figure 6 and Figure 7. It can be seen from the figure that, every time the environment changes, the curve of CCEA-DVC is the lowest or close to the lowest for most problems, which indicates that CCEA-DVC can quickly respond to environmental changes. For FDA1, dMOP2, and dMOP3, the IGD value of CCEA-DVC is sometimes bigger than that of PPS when the time window is bigger than 30. This may
be because the AR prediction model adopted in PPS needs to be trained with considerable accumulation of historical information. However, at the early stage of the environmental change, the information is not enough for PPS to learn the pattern of the environmental change. When the pattern is learned by PPS, it can predict the whole population accurately, thereby performing best at the latter stage of environmental change. For FDA3, in which environmental changes shift the POS and affect the density of points on the POF, the evolution curve of CCEA-DVC is the lowest in the whole stage of evaluation, proving that the special operations carried out on diversity variables are effective in maintaining population diversity.

In most JY problems, CCEA-DVC can achieve excellent performance. Especially at the early stage of environmental changes, CCEA-DVC can quickly converge to the POF. It is reflected in the figure that in the first 10 time windows, the IGD curve of CCEA is significantly lower than that of other algorithms. It indicates that the new cooperative coevolution mechanism proposed in this paper is efficient in handling DMOPs. But for the problem that the nonlinear relationships between decision variables change over time such as in JY3, these five algorithms all do not perform well in such problems. Since CCEA-DVC does not take into account the internal connection of decision variables, the performance of CCEA-DVC is similar to FPS and PPS, and worse than SGEA and DNSGA-II. For JY6 and JY7, the evolution curve of CCEA-DVC is similar to that of SGEA, which shows that CCEA-DVC is competitive when dealing with multimodal problems.

6. Discussion

6.1. Influence of severity of change

To observe the impact of the severity of environmental change \((n_t)\) on the performance of the algorithm on different problems, we conducted an experimental comparison on the three test suites: FDA, dMOP and JY, fixing the environmental change frequency \((\tau_t)\) to 15, and set \(n_t\) to 5, 15 and 20.
Figure 5: Evolution curves of average IGD values for FDA1∼FDA3 and dMOP problems with $n_t = 10$ and $\tau_t = 10$
Figure 6: Evolution curves of average IGD values for JY1~JY6 problems with $n_t = 10$ and $\tau_t = 10$
Table 5: Mean and SD of IGD indicator obtained by five algorithms (n_t changes).

<table>
<thead>
<tr>
<th>Prob. (n_t, n_o)</th>
<th>DNSGA-II</th>
<th>PPS</th>
<th>SGEA</th>
<th>CCEA-DVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5, 5)</td>
<td>1.031e+1(1.433e-3)</td>
<td>5.280e-1(1.206e-4)</td>
<td>6.409e-2(2.576e-4)</td>
<td>2.580e-1(1.099e-3)</td>
</tr>
<tr>
<td>(5, 10)</td>
<td>1.037e+1(1.437e-3)</td>
<td>5.285e-1(1.209e-4)</td>
<td>6.420e-2(2.578e-4)</td>
<td>2.584e-1(1.102e-3)</td>
</tr>
<tr>
<td>(5, 20)</td>
<td>1.039e+1(1.439e-3)</td>
<td>5.287e-1(1.211e-4)</td>
<td>6.423e-2(2.580e-4)</td>
<td>2.587e-1(1.105e-3)</td>
</tr>
<tr>
<td>(5, 50)</td>
<td>1.041e+1(1.441e-3)</td>
<td>5.300e-1(1.213e-4)</td>
<td>6.430e-2(2.583e-4)</td>
<td>2.590e-1(1.108e-3)</td>
</tr>
<tr>
<td>(5, 100)</td>
<td>1.043e+1(1.443e-3)</td>
<td>5.310e-1(1.214e-4)</td>
<td>6.435e-2(2.586e-4)</td>
<td>2.593e-1(1.111e-3)</td>
</tr>
<tr>
<td>(5, 500)</td>
<td>1.045e+1(1.445e-3)</td>
<td>5.320e-1(1.215e-4)</td>
<td>6.440e-2(2.589e-4)</td>
<td>2.596e-1(1.114e-3)</td>
</tr>
<tr>
<td>(5, 1000)</td>
<td>1.047e+1(1.447e-3)</td>
<td>5.330e-1(1.216e-4)</td>
<td>6.445e-2(2.592e-4)</td>
<td>2.599e-1(1.117e-3)</td>
</tr>
</tbody>
</table>

† and ‡ indicate CCEA-DVC performed significantly better than and equivalently to the corresponding algorithm respectively.
The experimental results of the five algorithms are presented in Table 5 in the form of IGD values. The influence of the severity of environmental change on the algorithm can be seen from the table; thus the sensitivity of the algorithm to \( n_t \) can be analyzed. As can be seen from the table, as \( n_t \) becomes larger, the smaller the severity of environmental change, and the better the performance of the algorithms. This shows that for dynamic algorithms, the severity of environmental change is an important factor that affects the performance of
the algorithms. In general, CCEA-DVC can perform well on most problems and outperformed other algorithms under different severities of change. It was slightly worse than SGEA on FDA4 and JY4, and worse than DNSGA-II on JY3. For JY3, CCEA-DVC had difficulty converging, so the performance fluctuation is not large, and DNSGA-II can improve the effect a lot after $n_t$ changes from 10 to 20, which shows that DNSGA-II is competitive in dealing with such problems. JY4, whose POF is composed of many fragments, is difficult for algorithms to converge. From the table, we can see that the performance of the five algorithms on JY4 basically does not change after $n_t$ changes from 10 to 20, which means that these five algorithms need a better way to handle the problems with a disconnected POF.

6.2. Study of different components of CCEA-DVC

This section studies the effect of different components of CCEA-DVC. CCEA-DVC has three key components. These are the special operations on diversity variables, the hybrid of the prediction and memory strategies for change response, and the cooperative coevolution for the evolution process. To deeply examine the role that each component plays in dynamic optimization, the original CCEA-DVC is adopted into three variants. The first variant (CCEA-DVC-S1) does not conduct special operations on the diversity variables. Instead, they are seen as the normal variables and all the variables are optimized like the traditional cooperative coevolution. The second variant (CCEA-DVC-S2) uses the random reinitialization to replace the dynamic response strategy presented in this paper. CCEA-DVC-S3 is another modification of CCEA-DVC, in which the cooperative coevolution mechanism is discarded and all the individuals are optimized in $DS$ (Algorithm 5). These three variants were compared with the original CCEA-DVC and the change frequency ($\tau_t$) and severity of change ($n_t$) are set to 10 and 10, respectively. The statistical results are shown in Table 6 including the average and standard deviation values of three indicators. The Wilcoxon rank-sum [31] is set to the 0.05 significance level.
Table 6: Mean and SD of IGD indicator obtained by CCEA-DVC and three variants of it.

<table>
<thead>
<tr>
<th>Prob. Indicator</th>
<th>CCEA-DVC-S1</th>
<th>CCEA-DVC-S2</th>
<th>CCEA-DVC-S3</th>
<th>CCEA-DVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDA1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GD</td>
<td>5.836e-2</td>
<td>6.668e-2</td>
<td>7.196e-2</td>
<td>6.223e-1</td>
</tr>
<tr>
<td>IGD</td>
<td>6.668e-2</td>
<td>7.196e-2</td>
<td>6.223e-1</td>
<td>6.223e-1</td>
</tr>
<tr>
<td>HV</td>
<td>1.391e-1</td>
<td>2.350e-2</td>
<td>3.589e-2</td>
<td>7.065e-1</td>
</tr>
<tr>
<td>FDA3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HV</td>
<td>6.223e-1</td>
<td>2.350e-2</td>
<td>3.589e-2</td>
<td>7.065e-1</td>
</tr>
<tr>
<td>dMOP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GD</td>
<td>4.438e-2</td>
<td>3.798e-2</td>
<td>2.164e-2</td>
<td>7.196e-2</td>
</tr>
<tr>
<td>IGD</td>
<td>3.798e-2</td>
<td>3.798e-2</td>
<td>3.798e-2</td>
<td>3.798e-2</td>
</tr>
<tr>
<td>HV</td>
<td>1.381e-1</td>
<td>7.030e-2</td>
<td>5.073e-2</td>
<td>9.95e-3</td>
</tr>
<tr>
<td>JY2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GD</td>
<td>1.484e-1</td>
<td>1.185e-1</td>
<td>2.164e-1</td>
<td>7.196e-2</td>
</tr>
<tr>
<td>IGD</td>
<td>1.185e-1</td>
<td>1.185e-1</td>
<td>2.164e-1</td>
<td>7.196e-2</td>
</tr>
<tr>
<td>HV</td>
<td>2.797e-1</td>
<td>2.797e-1</td>
<td>2.797e-1</td>
<td>2.797e-1</td>
</tr>
<tr>
<td>JY4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GD</td>
<td>3.188e+0</td>
<td>3.188e+0</td>
<td>3.188e+0</td>
<td>3.188e+0</td>
</tr>
<tr>
<td>HV</td>
<td>1.901e-1</td>
<td>1.901e-1</td>
<td>1.901e-1</td>
<td>1.901e-1</td>
</tr>
<tr>
<td>JY6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GD</td>
<td>4.371e+0</td>
<td>2.263e+0</td>
<td>1.901e+0</td>
<td>1.901e+0</td>
</tr>
<tr>
<td>IGD</td>
<td>2.263e+0</td>
<td>2.263e+0</td>
<td>1.901e+0</td>
<td>1.901e+0</td>
</tr>
<tr>
<td>HV</td>
<td>1.901e-1</td>
<td>1.901e-1</td>
<td>1.901e-1</td>
<td>1.901e-1</td>
</tr>
<tr>
<td>JY8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GD</td>
<td>1.850e-2</td>
<td>2.431e-2</td>
<td>1.382e-2</td>
<td>1.382e-2</td>
</tr>
<tr>
<td>IGD</td>
<td>2.431e-2</td>
<td>2.431e-2</td>
<td>1.382e-2</td>
<td>1.382e-2</td>
</tr>
<tr>
<td>HV</td>
<td>1.382e-2</td>
<td>1.382e-2</td>
<td>1.382e-2</td>
<td>1.382e-2</td>
</tr>
</tbody>
</table>

‡ and † indicate CCEA-DVC performed significantly better than and equivalently to the corresponding algorithm respectively.

As can be seen from Table 6, CCEA-DVC performed significantly better than the three variants on most problems including FDA3, dMOP1, JY2, JY4, and JY8, indicating that each component is essential and crucial for the excellent performance of CCEA-DVC. For FDA1, CCEA-DVC-S2 and CCEA-DVC obtained considerably smaller GD, IGD and HVD values, indicating that the presented algorithm can solve this problem very well even without a prediction strategy. For FDA3, CCEA-DVC-S1 and CCEA-DVC-S3 performed slightly worse than CCEA-DVC while better than CCEA-DVC-S2, which means that the diversity strategy used in change response is effective in handling the problem with diversity loss. For JY6, CCEA-DVC-S2 and CCEA-DVC significantly performed better than the others in IGD value, and CCEA-DVC-S2 performed best in IGD and HVD values, the reason may be that in the proposed cooperative coevolution method, the subpopulation which focuses on population con-
vergence may be combined with the subpopulation which evolves in the global search space. Therefore, this method is capable of handling the multimodal problems like JY6. When it comes to JY8, the difference between CCEA-DVC-S3 and CCEA-DVC in terms of GD and IGD is small and significantly better than the other two algorithms. One possible explanation is that the cooperative coevolution process can enhance the convergence of the population. From the above analysis, we can conclude that the three key components are indivisible for CCEA-DVC and contribute to population convergence or diversity.

7. Conclusions and future work

To handle dynamic multiobjective problems, this paper has proposed a decision variable classification-based cooperative coevolutionary algorithm. According to the different characteristics of the decision variables mapped to the objective space, this algorithm performs different operations on different types of variables in the static evolution process and change response. If a change is detected, CCEA-DVC conducts a simple prediction strategy on convergence variables. To obtain population diversity and compensate for the diversity loss caused by environmental change, the diversity variables of distributed individuals will remain, and those of another part of individuals are generated in the unsearched space. Otherwise, a generation cycle of static cooperative coevolution optimization is executed, in which the subpopulations optimizing convergence variables may be merged into the subpopulation which evolve in the global search space to help these individuals jump out of the local optimum. CCEA-DVC has been compared with four popular DMOEAs on 16 DMOPs with different dynamic characteristics and difficulties. Experimental studies show that CCEA-DVC is effective in handling DMOPs, especially for problems with a significant diversity loss caused by environmental change, as well as multimodal problems.

Although CCEA-DVC demonstrates an encouraging performance on the test instances, we also noticed its weakness on DMOPs with strong variable linkages.
and problems with a disconnected POF. It is our future work to design methods to handle these kinds of DMOPs. Moreover, CCEA-DVC needs to be examined on other kinds of DMOPs, such as problems with changing number of dimensions or problems which change irregularly. Additionally, the new performance indicators should be designed to assess the performance of algorithms accurately and comprehensively.

**CRediT authorship contribution statement**

**Huipeng Xie:** Conceptualization, Data curation, Software, Writing - original draft. **Juan Zou:** Supervision, Methodology. **Shengxiang Yang:** Supervision, Methodology, Writing - review & editing. **Jinhua Zheng:** Supervision, Investigation, Writing - review & editing. **Junwei Ou:** Software, Validation, Visualization. **Junwei Ou:** Software, Validation.

**Declaration of Competing Interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgement**

This work was supported by the research projects: the National Natural Science Foundation of China under Grant Nos. 61502408, 61673331, 61772178 and 61403326, the postgraduate research and innovation Project of Hunan Province under Grant No. XDCX2019B057.

**References**


