Information Sciences 607 (2022) 278–296

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Cooperative co-evolutionary algorithm for multi-objective optimization problems with changing decision variables

Biao Xu [a,b,c], Dunwei Gong [d,*], Yong Zhang [d], Shengxiang Yang [e], Ling Wang [f], Zhun Fan [a,c], Yonggang Zhang [b]

[a] Department of Electronic Engineering, Shantou University, Shantou 515063, China
[b] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[c] Key Laboratory of Digital Signal and Image Processing of Guangdong Province, Shantou 515063, China
[d] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China
[e] Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK
[f] Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 7 October 2021
Received in revised form 29 May 2022
Accepted 31 May 2022
Available online 6 June 2022

Keywords:
Dynamic optimization
Multi-objective optimization
Co-evolutionary algorithm
Grouping
Correlation

ABSTRACT

Multi-objective optimization problems (MOPs) with changing decision variables exist in the actual industrial production and daily life, which have changing Pareto sets and complex relations among decision variables and are difficult to solve. In this study, we present a cooperative co-evolutionary algorithm by dynamically grouping decision variables to effectively tackle MOPs with changing decision variables. In the presented algorithm, decision variables are grouped into a series of groups using maximum entropic epistasis (MEE) at first, with decision variables in different groups owning a weak dependency. Subsequently, a sub-population is generated to solve decision variables in each group with an existing multi-objective evolutionary algorithm (MOEA). Further, a complete solution including all the decision variables is achieved through the cooperation among sub-populations. Finally, when a decision variable is added or deleted from the existing problem, the grouping of decision variables is dynamically adjusted based on the correlation between the changed decision variable and existing groups. To verify the performance of the developed method, the presented method is compared with five popular methods by tackling eight benchmark optimization problems. The experimental results reveal that the presented method is superior in terms of diversity, convergence, and spread of solutions on most benchmark optimization problems.

1. Introduction

MOPs involve simultaneous optimization of two or more conflicting objectives [1,2]. In particular, an MOP with an increasing or decreasing dimension of decision variables over time is one with changing decision variables. A multi-period portfolio selection problem, for example, mainly addresses how to rationally allocate limited funds to various financial assets to balance the maximal return and minimal risk in a number of consecutive periods [3,4]. Owing to the changing market environment, the types and proportions of invested securities in each period are adjusted according to the current market environment. In the above process, investors are likely to hold on some of their existing assets while selling bad

* Corresponding author.

https://doi.org/10.1016/j.ins.2022.05.123
0020-0255/

assets and/or buying new ones with higher returns. If assets to be sold/bought are regarded as the decision variables when formulating the portfolio selection problem, it becomes a problem with changing decision variables. Specifically, it is an MOP with changing decision variables.

The changing of decision variables will result in a changing Pareto-optimal set (PS) for an optimization problem. Therefore, an MOP with changing decision variables can be regarded as a type of dynamic multi-objective optimization problem (DMOP) [2,5]. An approach of effectually tackling a DMOP needs to conquer difficulties raised by changing conditions within an optimization problem, such as following up a time-dependent Pareto front (PF) and offering solutions with good diversity. To this end, Deb et al. introduced diversity by randomly initializing a population or conducting the mutation operator on certain solutions chosen from the population [6]; Liang et al. incorporated a hybrid of memory and prediction strategies into a multi-objective evolutionary algorithm based on decomposition (MOEA/D) [7], in which a differential prediction is employed to relocate the population individuals in the new environment if a detected change is dissimilar to any historical changes, and a memory-based technique devised to predict the new locations of the population members is applied if a similar environment exists in the historical changes. In [8], a dynamic multi-objective evolutionary algorithm driven by inverse reinforcement learning is used to tackle the DMOPs. To accelerate the convergence and maintain the diversity of the evolutionary population, a Q-learning-based change response approach is considered to generate solutions in the promising regions; and [9] presented a multidirectional prediction strategy by clustering a population into several representative groups to improve the performance of evolutionary algorithms (EAs) in solving a DMOP. Recently, Li et al. provided a dynamic two-archive EA to tackle DMOPs with a changing number of objectives [10]. Two complementary co-evolving populations are simultaneously maintained to adaptively reconstruct their compositions once the environment changes and interact with each other via a mating selection mechanism.

However, all these methods have not considered the case where the number of decision variables increases or decreases over time. Although studies have been conducted on changing the number of decision variables for single-objective problems, such as [11,12], they consider different problems from those in our study. They aim to provide an approach for determining an optimal search dimension with a small population size when tackling a problem that has an infinite search dimension. To this end, an evolutionary algorithm is first employed to address an optimization problem with a small number of decision variables. Then, the number of decision variables is gradually increased during the optimization. The increase in search dimension is continued until performance cannot be improved within a certain number of generations after the dimension increases. In this case, the search dimension is decreased by one and remains unchanged. Finally, an optimal search dimension for a small population is provided. This indicates that the increasing or decreasing number of decision variables is only considered a strategy for solving the problem, which is not the characteristic of the problem. For a large-scale MOP with changing decision variables, running the current optimization method based on a new population is inefficient when a change in the number of decision variables is detected. Furthermore, producing a complete set of solutions to a DMOP with a large scale of decision variables using previous methods often results in an insurmountable computational complexity, suggesting their inefficiency in tackling an MOP with changing decision variables. Therefore, seeking appropriate methods for an MOP with changing decision variables is critical, and it is the key focus of this study.

It has been shown that co-evolutionary mechanisms can significantlyimprove the efficiency of the optimization process [2,13]. Because cooperative co-evolutionary algorithms (CCEAs) can significantly shrink the searching space of a subpopulation, they are efficient when solving a single-objective large-scale optimization problem [14–16]. In addition, CCEAs have been employed in conjunction with other strategies to address MOPs and DMOPs. For example, Li et al. developed a systematic way of incorporating the decision maker's preference information into the decomposition-based evolutionary multiobjective optimization methods [17]. Therefore, the search process is steered toward the region of interest for the decision maker directly or interactively. Moreover, to help decision makers identify solution(s) of interest from a given set of trade-off solutions in an MOP, they presented a simple and effective knee point identification method from a decomposition perspective [18]. The basic idea is to sequentially validate whether a solution is a knee point or not by comparing its localized trade-off utility with others within its neighborhood characterized from a decomposition perspective. However, they decomposed the problems in the objective space, which is difficult to adapt to problems with changing decision variables. [19] presented a distributed CCEA by exploiting the inherent parallelism of cooperative co-evolution (CC), which divides an MOP into several sub-problems based on the decision variables; each sub-problem contains only one decision variable and is optimized by a sub-population. [20] proposed a dynamic competitive-cooperative co-evolutionary algorithm to address DMOPs, where all the decision variables are adaptively classified into several groups and random competitors are utilized to track the moving optima. Two approaches to large-scale multi-objective optimization were introduced in [21,22]. One is to solve an MOP with many decision variables, called an evolutionary algorithm for large-scale many-objective optimization (LMEA), which divides the decision variables into distance- and diversity-related groups using a clustering approach [21]. The other is an MOEA based on decision variable analyses (MOEA/DVAs) for large-scale MOPs [22], which groups the decision variables according to the contribution of a decision variable to convergence (i.e., the distance to the PF), diversity, or both. However, many function evaluations are consumed before the optimization, especially for an optimization problem with numerous decision variables. Furthermore, these studies can only provide the correlation between decision variables, and are mainly suitable for decomposing decision variables of an optimization problem with separable objectives.

In recent years, researchers have proposed several novel grouping methods, such as differential grouping (DG), an improved variant of DG (DG2), global DG (GDG), and recursive DG (RDG). The DG [23] method identifies the interaction between decision variables by detecting the fitness changes when perturbing the decision variables. If the fitness change

induced by perturbing decision variable x_i varies for different value of x_j, x_i and x_j interact. However, DG is sensitive to the value of ε which is a parameter of DG used for determining whether two variables are nonseparable. To overcome this defect, DG2 [24] is proposed to adapt the value of ε to the objective value of a problem and identify the complete variable interaction matrix. Thus, the accuracy in identifying the interrelationship is improved. However, it may not need the entire variable interaction matrix to identify the connected subcomponents. In theGDG [25] method, the same technique as that used in DG is employed to identify the pairwise interactions between decision variables. The variable interaction matrix, also know as the adjacency matrix of a graph, is calculated. Then, the depth-first search or breadth-first search is used to identify the connected components. Note that both the interacting and conditionally interacting decision variables are placed into one connected subcomponent. The computational cost of DG, DG2, and GDG for decomposing a D-dimensional problemis $O(D^2)$. To reduce the computational cost of the differential grouping methods, RDG examines the interrelationship between a pair of sets of variables but not a pair of variables [26–28]. If two sets of variables (X_1 and X_2) are interrelated, RDG divides X_2 into two equal-sized subsets and examines the interrelationship between X_1 and the two subsets. The computational complexity is $O(Dlog_2D)$ when RDG decomposes a D-dimensional problem in the above binary search fashion. Nevertheless, the mentioned grouping methods, DG, DG2, GDG, and RDG, which divide the decision variables offline, can not adapt to problems associatedwith decision variable change. They lack a mechanism for adjusting the groups of decision variables as the optimization problem varies.

Therefore, it is extremely challenging for CCEAs to tackle an MOP with changing decision variables. The challenges include the following: (1) how to examine the change in an environment, (2) how to adjust the groups of decision variables when the environmental change occurs, and (3) how to respond to the environmental change. Bearing these challenges in mind, we provided some preliminary results [29]. In [29], a framework of parallel cooperative co-evolution based on dynamically grouping decision variables is proposed. With the Spearman rank correlation (SRC) analysis based on samples obtained from the evolution of a population, the decision variables are partitioned into several groups, which dynamically adjust when a change occurs. However, the samples have a significant influence on the above grouping results, causing inaccurate grouping results and low reliability. Moreover, only the case of increasing decision variables is considered, which is not adequate.

In this study, we further extend the previous work. An MOP with one decision variable being increased or decreased at a time is considered. Note that if more than one decision variable is changed simultaneously, we can handle them individually. A CCEA is presented for tackling this optimization problem. In this study, a number of groups are gained at first based on the relation among decision variables. As the dimension of decision variables increased or decreases, the interaction matrix (IM) between a newly added decision variable or an old reduced decision variable and each group will be computed according to information offered by the population to adjust the grouping of decision variables. Additionally, a hybrid strategy is employed to initialize sub-populations as the dimension of decision variables varies.

Some new features different from the previous work are provided as follows.

(1) Employing a novel method to accurately group the decision variables.
(2) Proposing an improved strategy to respond the change of decision variables.
(3) Detailing the strategy of evaluating an individual of a sub-population.
(4) Employing four new metrics to reflect the performances of the presented algorithm, and extending the experiments to survey the influences of different strategies on the proposed algorithm, which are beneficial to enriching the experiments.

The remainder of this paper is structured as follows. A comprehensively review on the related work is provided in Section 2. Section 3 details the proposed CCEA based on dynamically grouping decision variables. The experimental results are reported and analyzed in Section 4. Finally, Section 6 concludes the whole study.

2. Related work

2.1. Characteristics of MOP with changing decision variables

The following multi-objective minimization problem with changing decision variables is considered:

$$\begin{aligned} &\min F(X(t)) = (f_1(X(t)), f_2(X(t)), \cdots, f_M(X(t))) \\ &\text{s.t.} \begin{cases} g_i(X(t)) \leqslant 0, & i = 1, 2, \cdots, q \\ h_j(X(t)) = 0, & j = 1, 2, \cdots, s \\ X(t) \in [X_{\min}(t), X_{\max}(t)] \end{cases} \end{aligned} \tag{1}$$

where $f(\cdot)$ refers to a set consisting of M objectives to be minimized, $X(t) = (\omega_1(t)x_1, \omega_2(t)x_2, \cdots, \omega_D(t)x_D)$ is the decision vector containing at most D decision variables with $\omega_k(t)$ being a control parameter ($\omega_k(t) = 0$ *or* $1, \omega_k(t) = 1$ presents that the k-th component, x_k, is one of thedecision variables of the optimization problem at time t, and $\omega_k(t) = 0$ otherwise). $g_i(\cdot) \leqslant 0$ and $h_j(\cdot) = 0$ represent the i-th inequality and the j-th equality constraints, respectively.

In problem (1), the dimension of decision variables will increase or decrease because the value of $\omega_k(t)$ changes along with the environment. If the dimension of thedecision variables remains unchanged as parameter t changes, the problem

will be considered a traditional constant DMOP [6]. For the above two types of optimization problems, their PS and PF will change as a result of the environmental change. There are twofold differences between them. (1) The number of decision variables of a traditional DMOP remains unchanged, whereas the MOP discussed in this study has changing decision variables, and the fluctuation in the number of decision variables causes a change in the relationship among decision variables. (2) DMOPs have four types, i.e., Types I, II, IIII and IV [2,5]. However, for an MOP with changing decision variables, its true PS always varies with the varying dimension of decision variables, that is, Type I and Type II. More precisely, the true PS changes in the following two ways. First, similar to the true PS in Type III and Type IV,the true value of each original decision variable does not vary, and the true PS varies because of the additional decision variable, denoted as Type I(A) or Type II(A). Second, the true value of at least one original decision variable varies as the optimization problem varies, similar to the true PS in Type I and Type II, denoted as Type I(B) or Type II(B). Compared with the traditional DMOP, the PS change of the problem considered here is more complicated.

Therefore, it is hard for an algorithm to quickly track its optimal solutions because its PS varies with the dimension of the decision variables.

2.2. Challenges of MOP with changing decision variables

As the dimension of thedecision variables increase, the optimal solutions of the original MOP are no longer optimal for the new optimal problem. This is because the dimension of the optimal solutions of the original problem is insufficient and infeasible for the new problem. More importantly, the optimal solutions of the original MOP change as the problem varies (i.e., Type I(B) and Type II(B)), thereby causing the optimal solutions of the original problem to be unsuitable for the new problem. When addressing the multi-period portfolio selection problem with limited funds, the investment proportion of the original ones will inevitably decrease when a new asset is purchased, that the original portfolio needs to be adjusted.

In contrast, when the number of decision variables decreases, the optimal solutions to the original problem may change, resulting in poor convergence of the original problem. For example, in the multi-period portfolio selection problem with limited funds, it is inevitable to increase the investment proportion of holdings after selling non-performing assets. It can be noted that the original investment portfolio is not optimal for the new problem (i.e., the solution has poor convergence performance).

2.3. Maximal information coefficient (MIC)

MIC computes the mutual information (MI) [30] between two variables at all sorts of scales and locates the greatest possible MI at any scale. Let E mean a set of ordered pairs, $\{(x_i, y_i), \quad i = 1, \ldots, N\}, G$ represent an $m \times n$ grid covering E. That represents dimensions x and y are divided into m and n intervals, respectively. The probability density function of a grid cell is proportional to the number of data points inside that cell. The characteristic matrix $M(E)_{m,n}$ means the highest normalized MI of E with the $m \times n$ partition. Its definition is provided as follows.

$$M(E)_{m,n} = \frac{\max(\mathrm{MI})}{\log \min(m, n)} \tag{2}$$

where max(MI) means the maximum MI of E by all possible $m \times n$ partitions. The following is the definition for MIC of a set E.

$$\mathrm{MIC}(E) = \max_{0<mn<B(N)} \{M(E)_{m,n}\} \tag{3}$$

where N refers to the size of sample, and $B(N) = N^{0.6}$ was given by Reshef et al. [31]. MIC(E) means the maximal value in $M(E)$ subjected to $0 < mn < B(N)$.

3. The proposed algorithm

In this section, we present a CCEA based on dynamically classifying decision variables to efficiently handle an MOP with changing decision variables. In the proposed method, we first divide all the decision variables into several groups. Then, we adopt strategies to dynamically group decision variables and respond to the change when initializing sub-populations when the decision variables change during the evolution. In addition, we present a clustering-based method of evaluating individuals in a sub-population through cooperative co-evolution.

3.1. The Framework of the proposed CCEA

In this subsection, we present a cooperative co-evolutionary framework by combining the presented strategies into a conventional MOEA. In the presented framework, the decision variables of an optimization problem are first classified into a number of groups with decision variables in different groups owning weak dependency based on the strategy proposed in SubSection 3.2. Then, a sub-population is employed to solve decision variables in every group. Subsequently, during the cooperative co-evolution of sub-populations, a complete solution is formed using the strategy presented in SubSection 3.5, and its objective value is regarded as the fitness of an individual to be evaluated. The obtained non-dominated solu-

tions are stored in the archive, $A(t)$. If the optimization problem changes in the number of decision variables, the method of dynamically classifying decision variables and the mechanism of re-initialing sub-populations presented in SubSection 3.4 will be accordingly employed. The above process is repeated until a termination criterion set in advance is satisfied, and the complete non-dominated solutions in $A(t)$ are obtained. The pseudo code of the proposed framework is provided in Algorithm 1.

Algorithm 1: Pseudocode of the proposed framework

```
Input: the evolutionary generations, Gen;
Output: the archive, A(t).
Grouping the decision variables (Subsection 3.2);
t←0, Gen←0;
Initialize sub-populations;
while the stop criterion is not satisfied do
    Change examination (Subsection 3.3);
    if the change happens then
        t←t+1;
        Response to the problem change (Subsection 3.4);
    end
    Cooperatively co-evolve sub-populations with a pre-existing MOEA (Subsection 3.5);
    Choose non-dominated solutions and save them in A(t);
    Gen←Gen+1;
end
return A(t)
```

In Algorithm 1, the existing MOEA can be simply replaced by NSGA-II [32], SPEA2 [33], and MOPSO [34,35], with the purpose of the resulted algorithms having improved performance in a variety of aspects.

3.2. Forming the initial groups of decision variables

It is crucial to divide decision variables into a number of groups when tackling a complex optimization problem using a CCEA. The traditional grouping strategy based on the uniform partition method [36] does not consider the linkage relationship between decision variables, and hence is inefficient when tackling a complex optimization problem. Clustering analysis usually partitions the decision variables into a number of groups based on the Pearson correlation coefficient (PCC) [37] and SRC [29]. PCC is incapable of reflecting nonlinear relationships, and SRC can only capture increasing or decreasing relationships between decision variables. The groups obtained by them are not accurate because they both classify the decision variables based on the samples obtained from evolving populations, and different samples will result in different groups. Therefore, we attempt to classify the decision variables replacedbased onaccording to the MEE, which is more reliable and accurate than other grouping methods, such as PCC and SRC [38].

3.2.1. Measuring variable interaction with MIC

MEE classifies the decision variables by calculating the MIC. The process consists of the following two steps.

Step 1 Identifying direct interactions between decision variables according to the following Proposition 1 proposed by [38].

Proposition 1. *Let $f:\mathbb{R}^d\to\mathbb{R}$ be a differentiable function. If $\lim_{N\to\infty}\mathrm{MIC}((\partial f/\partial x_i),x_j)=\lim_{N\to\infty}\mathrm{MIC}(E)=\lim_{N\to\infty}\max_{0<mn<B(N)}\{M(E)_{m,n}\}=1$, x_i and x_j interact directly; if $\lim_{N\to\infty}\mathrm{MIC}((\partial f/\partial x_i),x_j)=\lim_{N\to\infty}\mathrm{MIC}(E)=\lim_{N\to\infty}\max_{0<mn<B(N)}\{M(E)_{m,n}\}=0$, x_i and x_j interact indirectly or are independent.*

For Proposition 1, N means size of samples, and E is a set with N samples, which are randomly generated. It is able to identify the direct interaction between decision variable pairs, x_i and x_j, in a differentiable function, f. Specifically, for a continuous optimization problem with its objective of $f(x)$, $\partial f/\partial x_i$ can be approximated with $\frac{f(x_i+\delta x_i)-f(x_i)}{\delta x_i}$. According to the above proposition, Sun et al. provided an information matrix with direct interactions, IM_d, by calculating a interaction degree, MIC [38]. If $\mathrm{MIC}((\partial f/\partial x_i),x_j)$ is greater than a threshold, α, then the corresponding entries of $\mathrm{IM}_d(i,j)$ will be set to 1, which means that x_i and x_j directly interact with each other. Otherwise, $\mathrm{IM}_d(i,j)=0$.

Step 2 Identifying indirect interactions between decision variables. This stage starts by constructing an interaction graph according to IM_d: setting each decision variable x_i as a vertex, i; connecting vertices i and j if $\mathrm{IM}_d(i,j)=1$. Then the breadth first search algorithm is utilized to determine strongly connected components. Ultimately, all the pairs of vertices i and j in

each identified strongly connected component are connected and the corresponding information matrix with direct or indirect interactions, IM(i,j), are set to 1, which means that the decision variables x_i and x_j interact each other and will be classified into the same group. IM$(i,j) = 0$ means that x_i and x_j are independent.

Let us consider the following function as an instance.

$$f(X) = (x_1 + x_2)^2 + (x_2 - x_3)^2 + x_4^2, X \in [-1,1]^4. \tag{4}$$

On the basis of the aforementioned method, we can obtain the IM_d and the IM, respectively, as follows.

$$\mathrm{IM}_d = \begin{bmatrix} - & 1 & 0 & 0 \\ 1 & - & 1 & 0 \\ 0 & 1 & - & 0 \\ 0 & 0 & 0 & - \end{bmatrix}, \mathrm{IM} = \begin{bmatrix} - & 1 & 1 & 0 \\ 1 & - & 1 & 0 \\ 1 & 1 & - & 0 \\ 0 & 0 & 0 & - \end{bmatrix}.$$

IM_d shows that both $\{x_1, x_2\}$ and $\{x_2, x_3\}$ interact with each other, and IM reports that x_1 and x_3 are indirectly interacted, which are linked by x_2. Therefore, the decision variables are classified into two groups: $\{x_1, x_2, x_3\}$ and $\{x_4\}$. Moreover, the total number of evaluations for each objective is $ND(D-1)$ at the first stage of MEE, where N means the sample size and D is the dimension of decision variables. In addition, the second stage of MEE does not consume any evaluations.

3.2.2. Forming the initial groups based on MEE

The above grouping method mainly focuses on single objective optimization problems, which cannot be applied directly to MOPs. Therefore, to form the initial groups of the decision variables for an MOP, we first calculate the IM_d^k for the kth $(k = 1, 2, \ldots, M)$ objective using the first stage of MEE. Then, all the IM_d^k are utilized to structure the IM_d using the following formula.

$$\mathrm{IM}_d(i,j) = \max_{k=1,2,\ldots,M.} (\mathrm{IM}_d^k(i,j)). \tag{5}$$

Finally, the decision variables are classified based on the IM deduced from the IM_d using the second stage of MEE.

The flow chart of forming initial groups of decision variables for an MOP with M objectives is depicted in Fig. 1.

To form the initial groups, we first calculate the IM_d^k value for the kth $(k = 1, 2, \ldots, M)$ objective using the first stage of MEE. Then, all the IM_d^k value are employed to form IM_d with formula (5). Finally, the initial groups of decision variables, $SX^1, SX^2, \cdots, SX^K$, are obtained based on the IM value deduced from the IM_d value in the second stage of MEE.

Algorithm 2 offers the pseudo code of the presented method of initial grouping the decision variables for an MOP.

Algorithm 2: Forming initial groups

Input: the size of objectives, M;
Output: the groups of decision variables, $SX^1, SX^2, \cdots, SX^K$.

```
for k = 1 to M do
    For the kth objective, f_k(X), execute the first stage of MEE and an information matrix, IM_d^k, is obtained;
end
Calculate IM_d by Eq. (5);
Classify the decision variables based on IM, deduced from IM_d, with the second stage of MEE;
return SX^1, SX^2, ..., SX^K
```

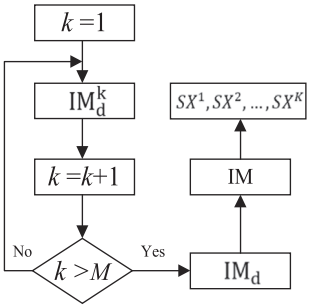

Fig. 1. Flow chart of forming initial groups of decision variables for an MOP.

To illustrate the process of Algorithm 2, consider the following two-objective problem as an example.

$$\begin{cases} f_1(X) = (x_1 + x_2)^2 + (x_2 - 1)^2 + x_3^2 + x_4^2 \\ f_2(X) = x_1^2 - (x_2 - 1)^2 + (x_2 - x_3)^2 - x_4^2 \end{cases} \tag{6}$$

According to the proposed method, we can obtain IM_d^1 and IM_d^2 from $f_1(X)$ and $f_2(X)$, respectively.

$$\mathrm{IM}_d^1 = \begin{bmatrix} - & 1 & 0 & 0 \\ 1 & - & 0 & 0 \\ 0 & 0 & - & 0 \\ 0 & 0 & 0 & - \end{bmatrix}, \quad \mathrm{IM}_d^2 = \begin{bmatrix} - & 0 & 0 & 0 \\ 0 & - & 1 & 0 \\ 0 & 1 & - & 0 \\ 0 & 0 & 0 & - \end{bmatrix}.$$

Furthermore,

$$\mathrm{IM}_d = \begin{bmatrix} - & 1 & 0 & 0 \\ 1 & - & 1 & 0 \\ 0 & 1 & - & 0 \\ 0 & 0 & 0 & - \end{bmatrix}, \quad \mathrm{IM} = \begin{bmatrix} - & 1 & 1 & 0 \\ 1 & - & 1 & 0 \\ 1 & 1 & - & 0 \\ 0 & 0 & 0 & - \end{bmatrix}.$$

At last, the decision variables are classified into two groups: $\{x_1, x_2, x_3\}$ and $\{x_4\}$ based on the IM.

The proposed grouping method has the following threefold characteristics. Firstly, the objectives are indirect utilized when generating the samples in MEE. Therefore, the interaction between decision variables obtained based on these samples is credible. In addition, the proposed grouping method can provide the interaction degree between decision variables. The larger the interaction degree between decision variables, the more interaction they have. Secondly, the proposed method arranges decision variables with interaction into the same group, which significantly reduces the size of each subcomponent. Finally, the proposed method is suitable for an optimization problem with both separable and non-separable objectives owing to its capability to identify the function correlated decision variables into the same group.

3.3. Change detection

The environment has a significant influence on solving an MOP with changing decision variables because previous non-dominated solutions may be dominated as the environment changes. Therefore, detecting whether or not the environment changing is essential. (1) To alleviate the negative influence of the randomness, two randomly selected solutions are reevaluated to detect whether a decision variable adds or not. The steps are explained in detail as follows. First, we assume a new environment and add a new decision variable to each randomly selected solution. The new decision variable is then assigned with two stochastic values in its feasible region, through which each selected solution corresponds to two new solutions. Finally, the fitness values of each selected solution and its corresponding new solutions are compared. If the fitness values are different, the environment is detected as changed, that is, a new decision variable is added. (2) Otherwise, the process of detecting whether a decision variable decreases or not is performed. For a randomly selected solution, we first disturb the value of one decision variable at a time. Then the fitness values of the selected solution and the disturbed solution are compared. If the fitness values are equal, (i.e. disturbing the value of the decision variable does not influence the fitness value) the disturbed decision variable is decreased.

3.4. Strategy for response decision variables change

3.4.1. Strategy for response groups of decision variables

Owing to the variability of decision variables, a DMOP contains not only a varying PS but also a varying relationship between decision variables. We assume that interaction between original decision variables remains unchanged. Under these circumstances, only the relation between the newly added or decreased variables, as well as the original decision variables are considered. If we keep the initial groups unchanged and take the newly added decision variable as a new group, a persistent part of building blocks with a close relation will be broken, resulting in an improper grouping. Further, regrouping the decision variables causes more computational complexity. Therefore, it is more suitable to re-investigate the relation among decision variables and further adjust groups when the problem changes. Accordingly, the burden of computation is reduced, and also the historical information of sub-populations is fully utilized. Based on the grouping results, CCEA can efficiently evolve in the subsequent generations, thereby reducing the time consumed in tackling the PS of a DMOP. In view of this, we propose a dynamically grouping strategy based on MEE in this subsection.

Let D be the current number of decision variables. When a decision variable, say x_{D+1} or x_r, is added or decreased to the optimization problem, the pseudo code of the dynamically grouping strategy is given in Algorithm 3.

Algorithm 3: Dynamically adjust the groups of decision variables

Input: the sizes of objectives and original decision variables, M and D, respectively; the information matrix, IM_d, in the previous environment; the threshold, α;
Output: the groups of decision variables, $SX^1, SX^2, \cdots, SX^K$.

```
if a decision variable, x_{D+1}, is added then
    Initialize an information matrix, IM'_d = zeros(D+1, D+1), and set IM'_d(1:D, 1:D) = IM_d;
    for k = 1 to M do
        for j=1 to D do
            if MIC((∂f/∂x_{D+1}), x_j) > α then
                IM'_d(D+1, j) = 1 and IM'_d(j, D+1) = 1;
            end
        end
    end
end
if a decision variable, x_r, is removed then
    Remove the rth row and column of IM_d, denoted as IM'_d;
end
Derive IM from IM'_d and classify the decision variables with the second stage of MEE;
return SX^1, SX^2, ..., SX^K
```

Based on the above analysis, we can conclude that the computation complexity of MEE is greater than that of DG, DG2, GDG and RDG, in the preliminary environment. However, it is more focused on the response speed of the algorithm as the environmental change in the DMOPs. In Algorithm 3, only the MIC between the newly added or decreased and the original decision variables are calculated to adjust the groups of decision variables. For each objective function, the maximum number of evaluations for dynamically adjusting groups of decision variables is $2ND$, which is $2/(D-1)$ of that of regrouping decision variables. Therefore, the computational burden of adjusting groups is lower than that of regrouping, which will accelerate the response speed of the environmental change. From this perspective, the proposed grouping method outperforms the existing methods, i.e., DG, DG2, GDG and RDG, which require regrouping the decision variable as the environment changes. Hence, the grouping method in this study is better adapted to the optimization problems with changing decision variables.

3.4.2. Strategy for response optimization sub-population

The PF of an optimization problem may change significantly with the change in decision variables. In this case, utilizing information provided by the population is beneficial in accelerating the convergence of an algorithm. When tackling DMOPs, researchers have proposed various methods that efficiently utilize historical information, such as the mutation strategy [2], and the approach of predicting the PF [9,39]. Nevertheless, these studies regard all the decision variables as a whole, and predict them using the same strategy. For an MOP with changing decision variables, the changed decision variable has different influences on different decision variables. Therefore, the algorithm will have a low efficiency if all the decision variables are predicted using the same strategy.

To this end, we present a method of responding to varying decision variables, which initializes sub-populations corresponding to different groups using different numbers of individuals in the archive. Specifically, for groups that are not affected by the newly added or decreased decision variable, we initialize the corresponding sub-populations by employing individuals in the archive with a high proportion, η, as these groups have a weak correlation with the changed decision variable. For other groups, we initialize their sub-populations by utilizing individuals in the archive with a low proportion, such as 0.5η. Moreover, we employ the following Gaussian perturbation operator to improve the diversity of each sub-population:

$$x_i(t+1) = a_i(t) + \varepsilon(t) \tag{7}$$

where $x_i(t+1)$ refers to the i-th individual at time $t+1$. $a_i(t)$ represents the i-th non-dominated solution randomly chosen from the archive, $A(t)$. $\varepsilon(t) \sim N(0, \sigma^2(t)I)$ is a Gaussian noise which is employed to promote the diversity of the population, I is an identity matrix, and $\sigma^2(t)$ is the variance, and $\sigma^2(t) = \frac{1}{n-1}\sum_{j=1}^{n}(a_j - \bar{a})^2, a_j \in A(t), \bar{a}$ and n are the mean and the size of the individuals in $A(t)$, respectively. It should be noted that an initial individual, $x_i(t+1)$, merely utilizes a part of information of $a_i(t)$ in (7), which corresponds to the decision variabes in the group to be optimized. For the other individuals in each of these sub-populations, they are re-initialized.

3.5. Evaluating Individuals of a sub-population

According to the approaches to choosing representative solutions, previous methods of evaluating individuals of a sub-population can be divided into the following two varieties. The first are random approaches, which randomly select representative solutions from either one or more sub-populations or the archive. Following that, the selected solutions are utilized to constitute one or more complete solutions, together with the individual to be evaluated. If the number of complete solution is more than one, the best one will be chosen, and its objective values will be regard as the fitness of the individual to be evaluated. Typical methods include the CCEA for dynamic multi-objective optimization problems [2,40]. Although the stochastic methods can maintain diverse sub-populations, they have a deteriorating performance in convergence due to the random selection of representative solutions.

The other category of approaches are preference-based methods. These approaches select representative solutions based on both Pareto domination and the distribution of individuals. For instance, [36] attempted to select a particle with a low rank and a small niche size as the representative solution. The fitness of an individual is calculated according to its contribution to the optimization problem and the dominance relation. For the preference-based methods, however, calculating the distribution of representatives is often computationally expensive.

To overcome the above limitations, we present a simple and efficient strategy for selecting representative solutions, in which the selection of a representative solution is based on clustering. In the proposed strategy, individuals in the other sub-populations are first divided into Kclusters with k-means clustering. Then, a representative individual is randomly selected from each cluster of the other sub-populations. Finally, for the individual to be evaluated, only Kcomplete solutions are formed by randomly combining it with the representative individuals in the other sub-populations. The objective values of the Kcomplete solutions are compared, and the best one is considered as the fitness of the individual to be evaluated. In addition, the obtained optimal solution is saved in the archive.

Figs. 2(a) and (b) demonstrate the course of choosing the representative individual for the case of 3 sub-populations. In Fig. 2(a), the decision variables are first classified into the following 3 groups: $SX^1 = (x_1^1, x_2^1, \cdots, x_{n_1}^1), SX^2 = (x_1^2, x_2^2, \cdots, x_{n_2}^2)$, and $SX^3 = (x_1^3, x_2^3, \cdots, x_{n_3}^3)$. Then, the sub-populations, P^1, P^2, andP^3, are utilized to optimized them, respectively.

Fig. 2(b) shows the method of evaluating an individual in P^1 when $K = 3$. Taking the i-th individual in P^1, i.e. p_{1i}, as an example, the individuals in $P^2$and $P^3$are first divided into $K = 3$clusters. Then, 3 representative individuals, $p_{2j_1}, p_{2j_2}$and p_{2j_3}, are randomly selected from each cluster of P^2, respectively. Similarly, $p_{3q_1}, p_{3q_2}$and p_{3q_3}, are selected from P^3. Finally, 3 complete solutions are formed by randomly combining p_{1i} with the representative individuals $p_{2j_1}, p_{2j_2}, p_{2j_3}$in $P^2$and $p_{3q_1}, p_{3q_2}, p_{3q_3}$in P^3, e.g., $s_1 = (p_{1i}, p_{2j_1}, p_{3q_3}), s_2 = (p_{1i}, p_{2j_2}, p_{3q_1}), s_3 = (p_{1i}, p_{2j_3}, p_{3q_2})$. The objective values of s_1, s_2, and s_3 are compared, and the best one, e.g., s_1, is regarded as the fitness of p_{1i}.

3.6. Complexity of the proposed algorithm

The major difference between NSGA-II and the proposed MEE-NSGA-II in each generation lies in grouping the decision variables, detecting the environmental change, responding the change, cooperatively co-evolving each sub-population, and forming the archive. Assume that there are q sub-populations with their size of P/q (P is the total size of the population) to cooperatively co-evolve when tackling an optimization problem with $M(\geq 2)$ objectives, N samples and D decision variables. The computational complexity associated with each of the above strategies is provided as follows:

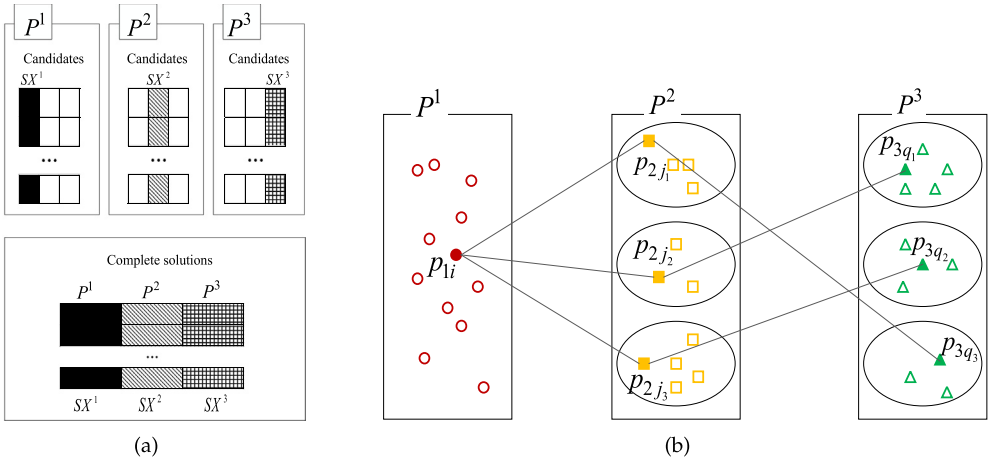

Fig. 2. Process of evaluating an individual in a sub-population.

(1) grouping the decision variables is $O(MND^2)$,
(2) detecting the environmental change is $O(D)$ in the worst case,
(3) responding to the change is $O(D)$,
(4) cooperatively co-evolving all sub-populations is $O(P)$, and
(5) forming the archive is $O(MP^2)$.

The number of decision variables and samples, D and N, is smaller than the size of the population, P. Therefore, the overall complexity of the proposed algorithm is $O(MP^2)$, equal to NSGA-II [32]. From this viewpoint, MEE-NSGA-II is computationally efficient.

4. Experimental study

To illustrate the superiority the presented algorithm, we carry out the following four groups of experiments. The first demonstrates the effects of different changing frequencies and grouping strategies. The MEE-based grouping method is compared to k-means clustering with feature measured by PCC and SRC, the uniform grouping [36], and no grouping methods (NSGA-II) [32] with different changing frequencies. The second evaluates the influences of the proposed strategy for selecting representative solutions. The third one incorporates the proposed strategies with two state-of-the-art algorithms i.e. SPEA2 [33] and MOPSO [41], and compare them with the two state-of-the-art algorithms. The last one compares the proposed algorithm, MEE-NSGA-II, with MOEA/DVA [22] and LMEA [21] on large-scale DMOPs.

4.1. Benchmark optimization problems

In this work, we mainly focus on a type of DMOPs in which the dynamics is caused by changing the number of decision variables. However, the dynamics of the problems mentioned in [2] is caused by environmental changes. If incorporating the two characteristics, changing the number of decision variables and environments, into a DMOP, it becomes more complex. ZDT and DTLZ are two static and continuous suites which can be extended to any number of decision variables. Therefore, we select two well-defined test suites, improved ZDT [42] and DTLZ [43], as the benchmark optimization problems, and denote them as DMOP1-DMOP8. To evaluate the performance of the proposed method, we take two- and three-objective optimization problems with various characteristics into consideration, such as linear, concave, non-concave, multi-modal, and disconnected (biased, or degenerate), which are summarized in Table 1 of the Supplementary material.

4.2. Performance metrics

We employ the generation distance (GD) [2,40], spacing (SP) [1,44], maximum spread (MS) [44], inverted generation distance (IGD) [45,46], and hyper-volume (HV) [39,47] to measure various performances of the proposed algorithm/strategy. To make them suitable for DMOPs, their improved versions, named as MGD, MSP, MMS, MIGD [2,39] and MHV [39], are treated as the performance metrics. Among them, the MGD metric means of the GD values in a number of time steps for each run, with its expression being as follows.

$$MGD = \frac{1}{|T|}\sum_{t\in T} GD(P^{t*}, P^t) \tag{8}$$

where T refers to a set of time steps for each run, with its cardinality being, $|T|$. P^{t*} represents a number of uniformly distributed Pareto optimal points in the true PF, and P^t is an approximation of PF. Similarly, we have the meaning of MSP, MMS, MIGD, and MHV.

Among these metrics, MGD and MSP measure the convergence and the diversity of a solution set, respectively. MGD = 0 represents the achieved optimal solutions are located at the true PF. MSP = 0 means the achieved solution set is homogeneous distribution.

MMS refers to how well the true PF is covered by the achieved PF, and a larger value of MS in the interval of [0, 1] reflects a better extent (or spread) of the optimal solutions. Both MIGD and MHV can measure the performances of an algorithm in convergence and diversity, and the larger (smaller) the value of MIGD (MHV), the worse the algorithm is.

When calculating the values of GD and IGD, a set of reference points with the uniform distribution on the PF is required. 5,000 reference points are employed for all the optimization problems. Moreover, $(z_1, z_2, \cdots, z_M)$ is a reference point to calculate HV, where z_i is the maximal value of the i-th objective function of the true PF.

4.3. Compared algorithms and parameter settings

Particularly, the proposed strategies are compatible with any type of population-based optimization algorithms. As a case study, five state-of-the-art MOEAs, NSGA-II, SPEA2, MOPSO, MOEA/DVA and LMEA, are chosen to verify our approaches. The proposed strategies, MEE-base grouping CCEA and dynamically grouping, are incorporated with NSGA-II, SPEA2, MOPSO,

called as MEE-NSGA-II, MEE-SPEA2 and MEE-MOPSO, respectively, for MOPs with decision variables changing one by one. The three new algorithms, MEE-NSGA-II, MEE-SPEA2 and MEE-MOPSO, are compared with NSGA-II, SPEA2, MOPSO, MOEA/DVA and LMEA in our experiments.

Additionally, to verify the superiority of the MEE-based grouping method, we compare MEE-NSGA-II with a cooperative co-evolutionary NSGA-II based on *k*-means clustering with feature measured by the Pearson correlation coefficient between decision variables, denoted as PCC-NSGA-II, and by the Spearman rank correlation coefficient, denoted as SCC-NSGA-II [29], and uniform grouping-based [36] cooperative co-evolutionary NSGA-II, denoted as UCC-NSGA-II.

These selected algorithms are popular and show good performance in static environments. Meanwhile, the grouping methods cover main categories mentioned in Section II for MOPs.

With regard to NSGA-II and SPEA2, the simulated binary crossover and the polynomial mutation operators are adopted with their distribution indexes of 20. The crossover and mutation probabilities are 0.9 and 0.1, respectively. The termination criterion is that the number of evaluations reaches to a value predefined in advance, i.e. 600,000 for all the algorithms. The size of each of the populations is set to 50 for all the compared algorithms. A complete solution is made up by the method proposed in SubSection 3.5 to evaluate an individual for all these algorithms. For *k*-means clustering, the number of clusters, *K*, i.e. the size of representative individuals, is set to 3 considering the computational complexity and the conclusion in [40]. α in Proposition 1 is set to 0.2 [38]. Besides, the parameter settings for the MEE- and SCC-based algorithms are the same as those in the original studies [29,38]. MOPSO and MEE-MOPSO take an inertia weight of 0.4 and a mutation rate of 0.5 [41].

For each optimization problem, $t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$, where n_t and τ_t are the severity and frequency of a change, respectively, and $\tau$is the maximal number of iterations. To investigate the influences of the change frequency on algorithms' ability in tracing dynamic environments, we only set $n_t = 1$ and let τ_t vary due to reasons given in Sections 1 and 4.3. Specifically, we set (n_t, τ_t)=(1, 15), (1, 20), and (1, 25), and τ =150, 200, and 250, respectively. The number of decision variables is set to 200 in the experiments except as otherwise noted. To study the influences of the strategy for selecting representative solutions, (n_t, τ_t) is set as (1, 25) and τ =250.

Each algorithm runs independently 30 times on each benchmark problem, and the algorithm is terminated when $t > 10$associate with the above settings of the frequency, showing that there are ten varies in total for every DMOP. The average and standard deviation of each metric over 30 runs are computed before the optimization problem varies. In addition, the Wilcoxon rank-sum test at the significant level of 0.05 is engaged.

4.4. Grouping accuracy of the proposed grouping method over time

To illustrate the grouping accuracy of the proposed grouping method over time, the situation, (n_t, τ_t)=(1, 25), five varies for the environmental change, the number of decision variables is 20, are taken as an example. *Recall* defined in [38] is used to measure the grouping accuracy over time, including the initial grouping and the adjustment during the process.

Table 1 lists the experimental results when the proposed grouping method is utilized to group the decision variables in the 8 benchmark functions. Then, the following concludes can be obtained. In the initial stage (t = 0), the grouping method obtains the maximum value of *Recall* on most of the benchmark functions except on DMOP6. Furthermore, it achieves best scores on DMOP1, DMOP2, DMOP3, and DMOP5 over time. The scores on DMOP4, DMOP6, DMOP7, and DMOP8 are little less than 1 at some moments, especially on DMOP6, DMOP7 and DMOP8. The cause might be the error of calculation. There are three objectives for DMOP6, DMOP7 and DMOP8, which will increase the complexity in grouping the decision variables. In summary, the proposed grouping method is able to accurately decompose the changing decision variables over time. Table 1.

4.5. Influences of different changing frequencies

Table 2 lists the mean and standard deviation values of MIGD and MHV achieved by MEE-NSGA-II and the other five compared algorithms with (n_t, τ_t)=(1, 15), (1, 20), and (1, 25). The following observations can be obtained.

(1) The MIGD values of the compared algorithms except NSGA-II become small on almost all the test problems with the increase of the changing frequency. In addition, NSGA-II has a great fluctuation in MIGD with various changing frequen-

Table 1
Grouping accuracy of the proposed grouping method on 8 benchmark functions in terms of *Recall* over time.

Fun.	0	1	2	3	4	5
DMOP1	1.00	1.00	1.00	1.00	1.00	1.00
DMOP2	1.00	1.00	1.00	1.00	1.00	1.00
DMOP3	1.00	1.00	1.00	1.00	1.00	1.00
DMOP4	1.00	1.00	1.00	0.98	0.99	1.00
DMOP5	1.00	1.00	1.00	1.00	1.00	1.00
DMOP6	0.98	0.96	0.99	0.95	0.95	0.96
DMOP7	1.00	0.98	0.95	0.96	0.98	0.98
DMOP8	1.00	1.00	0.99	0.98	0.99	1.00

Table 2
Performance comparisons of different grouping strategies with different change frequencies.

Problem	(n_t, τ_t)	*MIGD*					*MHV*				
		NSGA-II	PCC-NSGA-II	UCC-NSGA-II	SCC-NSGA-II	MEE-NSGA-II	NSGA-II	PCC-NSGA-II	UCC-NSGA-II	SCC-NSGA-II	MEE-NSGA-II
DMOP1	(1, 15)	0.0094	0.0079	0.0106	0.0078	**0.0073**	0.9027	0.9192	0.9019	0.9201	**0.9202**
		(0.0010)†	(0.0011)	(0.0017)†	(0.0010)	**(0.0011)**	(0.0954)†	(0.0989)	(0.0956)†	(0.0985)	**(0.0984)**
	(1, 20)	0.0087	0.0075	0.0097	0.0078	**0.0073**	0.9165	0.9194	0.9101	0.9197	**0.9202**
		(0.0038)†	(0.0008)	(0.0012)†	(0.0018)	**(0.0011)**	(0.0968)	(0.0995)	(0.0970)	(0.0991)	**(0.0986)**
	(1, 25)	0.0070	0.0073	0.0077	0.0068	**0.0062**	0.9697	0.9694	0.9683	0.9709	**0.9821**
		(0.0001)	(0.0007)	(0.0001)†	(0.0001)	**(0.0001)**	(0.0002)†	(0.0003)†	(0.0006)†	(0.0003)†	**(0.0002)**
DMOP2	(1, 15)	0.0075	0.0080	0.0097	**0.0068**	0.0069	0.2521	0.3208	0.2374	0.3255	**0.3258**
		(0.0014)	(0.0009)†	(0.0023)†	(**0.0006**)	(0.0008)	(0.0644)†	(0.0026)	(0.0817)†	(0.0017)	**(0.0014)**
	(1, 20)	0.0077	0.0074	0.0087	0.0066	**0.0064**	0.2981	0.3220	0.2906	**0.3258**	0.3251
		(0.0017)†	(0.0008)	(0.0032)†	(0.0006)	**(0.0005)**	(0.0259)†	(0.0021)	(0.0453)†	**(0.0016)**	(0.0018)
	(1, 25)	0.0069	0.0070	0.0074	0.0073	**0.0062**	0.3226	0.3243	0.3116	0.3239	**0.3251**
		(0.0001)	(0.0001)	(0.0002)†	(0.0001)†	**(0.0002)**	(0.0005)†	(0.0006)	(0.0006)†	(0.0001)	**(0.0003)**
DMOP3	(1, 15)	0.0107	0.0104	0.0116	0.0103	**0.0092**	0.7560	0.7713	0.7506	0.7718	**0.7765**
		(0.0028)†	(0.0010)	(0.0023)†	(0.0009)	**(0.0009)**	(0.0218)†	(0.0028)	(0.0302)†	(0.0025)	**(0.0025)**
	(1, 20)	0.0088	0.0091	0.0105	0.0102	**0.0090**	0.7712	0.7760	0.7688	0.7753	**0.7764**
		(0.0026)	(0.0008)	(0.0012)†	(0.0039)†	**(0.0007)**	(0.0078)	(0.0030)	(0.0105)	(0.0023)	**(0.0026)**
	(1, 25)	0.0090	0.0093	0.0095	0.0094	**0.0083**	0.7770	0.7760	0.7751	0.7760	**0.7786**
		(0.0001)†	(0.0002)†	(0.0002)†	(0.0002)†	(**0.0001**)	(0.0002)†	(0.0003)†	(0.0005)†	(0.0005)†	**(0.0003)**
DMOP4	(1, 15)	0.4904	0.3847	0.3831	0.3638	**0.3578**	0.3501	0.5450	0.5309	0.7158	**0.7200**
		(0.0388)†	(0.0461)†	(0.0442)†	(0.0401)	**(0.0469)**	(0.2346)†	(0.3029)†	(0.2272)†	(0.0969)	**(0.0986)**
	(1, 20)	0.4238	0.3524	0.3871	0.3591	**0.3509**	0.3891	0.5952	0.5842	0.6951	**0.7166**
		(0.0372)†	(0.0434)†	(0.0384)†	(0.0404)	**(0.0512)**	(0.2424)†	(0.2840)†	(0.2339)†	(0.1102)†	**(0.0969)**
	(1, 25)	0.4545	0.3769	0.3914	**0.2410**	0.4390	0.2739	0.5328	0.5805	**0.7675**	0.7611
		(0.0170)	(0.0330)	(0.0881)	**(0.0341)**†	(0.0410)	(0.0170)†	(0.0140)†	(0.0937)†	**(0.0190)**	(0.0002)
DMOP5	(1, 15)	0.0454	0.1130	0.0521	**0.0442**	0.0496	0.2138	0.2235	0.2101	**0.2638**	0.2635
		(0.0253)	(0.0255)†	(0.0240)†	(**0.0110**)	(0.0173)	(0.0431)†	(0.0775)†	(0.0487)†	**(0.0012)**	(0.0011)
	(1, 20)	0.4562	0.0585	**0.0417**	0.0496	0.0493	0.2150	0.2533	0.2216	0.2630	**0.2636**
		(0.0248)†	(0.0408)†	(**0.0197**)	(0.0174)†	(0.0176)†	(0.0622)†	(0.0271)	(0.0510)†	(0.0019)	**(0.0010)**
	(1, 25)	0.0422	0.0410	0.0415	0.0416	**0.0359**	0.2563	0.2575	0.2569	0.2580	**0.2671**
		(0.0039)†	(0.0002)†	(0.0001)†	(0.0001)†	**(0.0001)**	(0.0061)†	(0.0004)†	(0.0003)†	(0.0003)†	**(0.0002)**
DMOP6	(1, 15)	0.1933	0.2378	0.1944	0.1725	**0.1472**	0.0461	0.0280	0.0206	0.0452	**0.0469**
		(0.0449)	(0.1478)	(0.0712)	(0.0629)†	**(0.0274)**	(0.0094)	(0.0026)†	(0.0051)†	(0.0120)	**(0.0203)**
	(1, 20)	0.1767	0.2155	0.1821	0.1834	**0.1448**	0.0437	0.0310	0.0430	0.0412	**0.0471**
		(0.0468)†	(0.1292)†	(0.0510)†	(0.0646)†	**(0.0266)**	(0.0138)†	(0.0027)†	(0.0094)	(0.0227)†	**(0.0105)**
	(1, 25)	0.4115	0.2311	0.1367	0.1229	**0.1052**	0.0085	0.0461	0.0293	**0.0620**	0.0615
		(0.0210)†	(0.0260)†	(0.0280)†	(0.0130)	**(0.0101)**	(0.0008)†	(0.0008)†	(0.0019)†	**(0.0007)**	(0.0030)
DMOP7	(1, 15)	0.0891	0.0842	0.0993	0.0808	**0.0769**	0.5838	0.6054	0.5710	**0.6561**	0.6421
		(0.0100)†	(0.0076)†	(0.0106)†	(0.0084)†	**(0.0070)**	(0.2467)†	(0.2427)	(0.2548)†	**(0.2457)**	(0.2395)
	(1, 20)	0.0833	0.0809	0.0957	0.0812	**0.0768**	0.6030	0.6265	0.5934	0.6396	**0.6549**
		(0.0083)†	(0.0070)	(0.0102)†	(0.0081)†	**(0.0072)**	(0.2466)†	(0.2411)	(0.2401)†	(0.2465)	**(0.2471)**
	(1, 25)	0.0894	**0.0818**	0.0919	0.0883	0.0831	0.7207	0.7290	0.7377	0.7437	**0.7847**
		(0.0009)	**(0.0007)**	(0.0009)	(0.0011)	(0.0007)	(0.0110)†	(0.0072)†	(0.0134)†	(0.0026)†	**(0.0027)**
DMOP8	(1, 15)	0.5651	0.7467	0.9780	0.5640	**0.5471**	0.0062	0.0025	0.0083	**0.0742**	0.0604
		(0.0799)†	(0.0680)†	(0.0529)†	(0.3369)†	**(0.0909)**	(0.0017)†	(0.001)†	(0.0045)†	**(0.0098)**	(0.0154)†
	(1, 20)	0.5301	0.6978	0.9149	0.5276	**0.5137**	0.0061	0.0027	0.0093	0.0791	**0.0861**
		(0.0551)†	(0.0590)†	(0.0545)†	(0.3200)†	**(0.0738)**	(0.0185)†	(0.0011)†	(0.0080)†	(0.0158)†	**(0.0094)**
	(1, 25)	0.5762	0.6897	0.9070	0.5237	**0.5012**	0.0026	0.0034	0.0041	0.0489	**0.0781**
		(0.0680)†	(0.0410)†	(0.0190)†	(0.0270)†	**(0.0280)**	(0.0023)†	(0.0031)†	(0.0030)†	(0.0093)†	**(0.0081)**

cies. These indicate that the co-evolutionary algorithms have good performance in convergence and distribution with various changing frequencies.

(2) MEE-NSGA-II greatly improves the performance in convergence and distribution of NSGA-II on almost all test problems with various changing frequencies, suggesting that NSGA-II with the grouping strategy has good performance in convergence and distribution on the premise of tracing dynamic environments.

(3) For DMOP1, DMOP3, and DMOP8, MEE-NSGA-II achieves better performance in terms of the MIGD metric than the other co-evolutionary algorithms with various changing frequencies, especially, it is significantly superior to UCC-NSGA-II. For DMOP2, DMOP6, and DMOP7, MEE-NSGA-II obtains better performance in terms of MIGD in two out of three change frequencies than the compared algorithms, indicating that the proposed grouping strategy is more effective than its counterparts.

(4) For DMOP5 with all the frequencies of the environment change, MEE-NSGA-II has slightly worse performance for the MIGD metric than UCC- and SCC-NSGA-II.

From data on the right columns of Table 2, we can achieve similar observations for the MHV metric.

4.6. Influences of different grouping strategies

Furthermore, the situation, (n_t, τ_t) =(1, 25), is taken as an example to analyze the influences of different grouping strategies. The following observations can be achieved from Table 2, where data outside/in the bracket are the means/standard deviations of metrics, bold type ones are the greatest among these methods, and those labeled as '†' mean that results acquired between two exists are significantly different at the significant level of 0.05.

(1) For DMOP1 and DMOP2, with the help of the proposed grouping strategy, MEE-NSGA-II performs the best, followed by NSGA-II, SCC-NSGA-II, PCC-NSGA-II, and UCC-NSGA-II. Although there is no significant difference in terms of MIGD between MEE-NSGA-II and NSGA-II, NSGA-II obtains a lower MHV value. Additionally, for DMOP1 and DMOP2, NSGA-II significantly outperforms PCC-NSGA-II, and UCC-NSGA-II in terms of MIGD, indicating that an inappropriate classifying method is likely to deteriorate the performance of an algorithm.
(2) MEE-NSGA-II is significantly superior to the other four on DMOP3, DMOP5, and DMOP8. Taking DMOP8 as an example, MEE-NSGA-II has the MIGD value of 0.50120, which is better than NSGA-II(0.57620), PCC-NSGA-II(0.68970), UCC-NSGA-II(0.9070), and SCC-NSGA-II(0.5237), suggesting that MEE-NSGA-II has better performance in terms of distribution and convergence.
(3) For DMOP4, SCC-NSGA-II achieves a better value of MIGD than MEE-NSGA-II, PCC-NSGA-II, and UCC-NSGA-II. However, there is no significant difference between MEE-NSGA-II and NSGA-II, PCC-NSGA-II, and UCC-NSGA-II. MEE-NSGA-II has a larger value of MHV than NSGA-II and UCC-NSGA-II, and approximates that of SCC-NSGA-II.
(4) For DMOP6 which is a multi-modal optimization problem with plenty of local optima, MEE-NSGA-II achieves the best performance on the MIGD metric. Moreover, MEE-NSGA-II significantly outperforms NSGA-II, PCC-NSGA-II, and UCC-NSGA-II on MHV. Although the MHV value of SCC-NSGA-II, 0.06201, is bigger than that of MEE-NSGA-II, 0.06150, they have no significant difference. The true PF of DMOP7 changes as the number of decision variables increases or decreases. Hence, it is not easily tracked when the optimization problem varies. Although PCC-NSGA-II archives the lowest MIGD value, each pair of algorithms has no significant difference with respect to MIGD. Furthermore, MEE-NSGA-II has the biggest value of MHV. Therefore, the presented method can handle DMOP6 and DMOP7 with good capability.

Furthermore, Fig. 3 depicts the curves of the IGD and HV values when tackling DMOP1-DMOP8 with (n_t, τ_t) =(1, 25). We have the following observations from this figure.

(1) For DMOP1, DMOP2, and DMOP3, four CCEAs show the similar performance in convergence and diversity with NSGA-II. Nevertheless, the proposed algorithm obtains the minimal IGD values along with the change of the optimization problems, showing its excellence in tracking time-dependent PFs. In addition, the IGD values of the proposed algorithm have a less fluctuation than those of the others, which emphasizes its powerful stability.
(2) For DMOP4, its PS and PF change over time. For this problem, UCC-NSGA-II achieves the best values of IGD and HV. As the problem changes, the proposed algorithm shows the same good performance as UCC-NSGA-II, given the fact that their slight difference in terms of IGD and HV values. NSGA-II, nevertheless, performs poorly since it employs no grouping strategy, resulting in a low efficiency in tracking the true PF when the dimension of decision variables changes.
(3) For DMOP5, the proposed method is excel in robustness no matter how the optimization problem varies.
(4) Both DMOP6 and DMOP8 are three-objective optimization problems, with a great number of local optima. For these two problems, all the four CCEAs gain a better performance than NSGA-II in terms of HV, and the proposed algorithm performs best among them, with a smaller value of IGD than the others.
(5) DMOP7 is also a three-objective optimization problem, whose PS and PF change over time. When tackling this problem, MEE-NSGA-II achieves the best values of IGD and HV compared with SCC-NSGA-II, NSGA-II, and UCC-NSGA-II, and approximates to that of PCC-NSGA-II in terms of HV, suggesting its excellent capability in solving DMOPs.

As mentioned above, we can draw the following conclusions: (1) the proposed algorithm, MEE-NSGA-II, has significant advantages in convergence and distribution with the increase of the changing frequency, (2) grouping decision variables is necessary when solving complex optimization problems, and (3) appropriate grouping strategy is beneficial to improving CCEAs.

4.7. Influences of the strategy of selecting representative solutions

The proposed strategy of selecting representative solutions is compared with the random-based and preference-based ones in this subsection. The proposed algorithm is employed to solve DMOP1, DMOP2, DMOP6, and DMOP7 based on the above three strategies. MGD, MSP and MMS, which can specifically investigate the performance of an algorithm in a certain

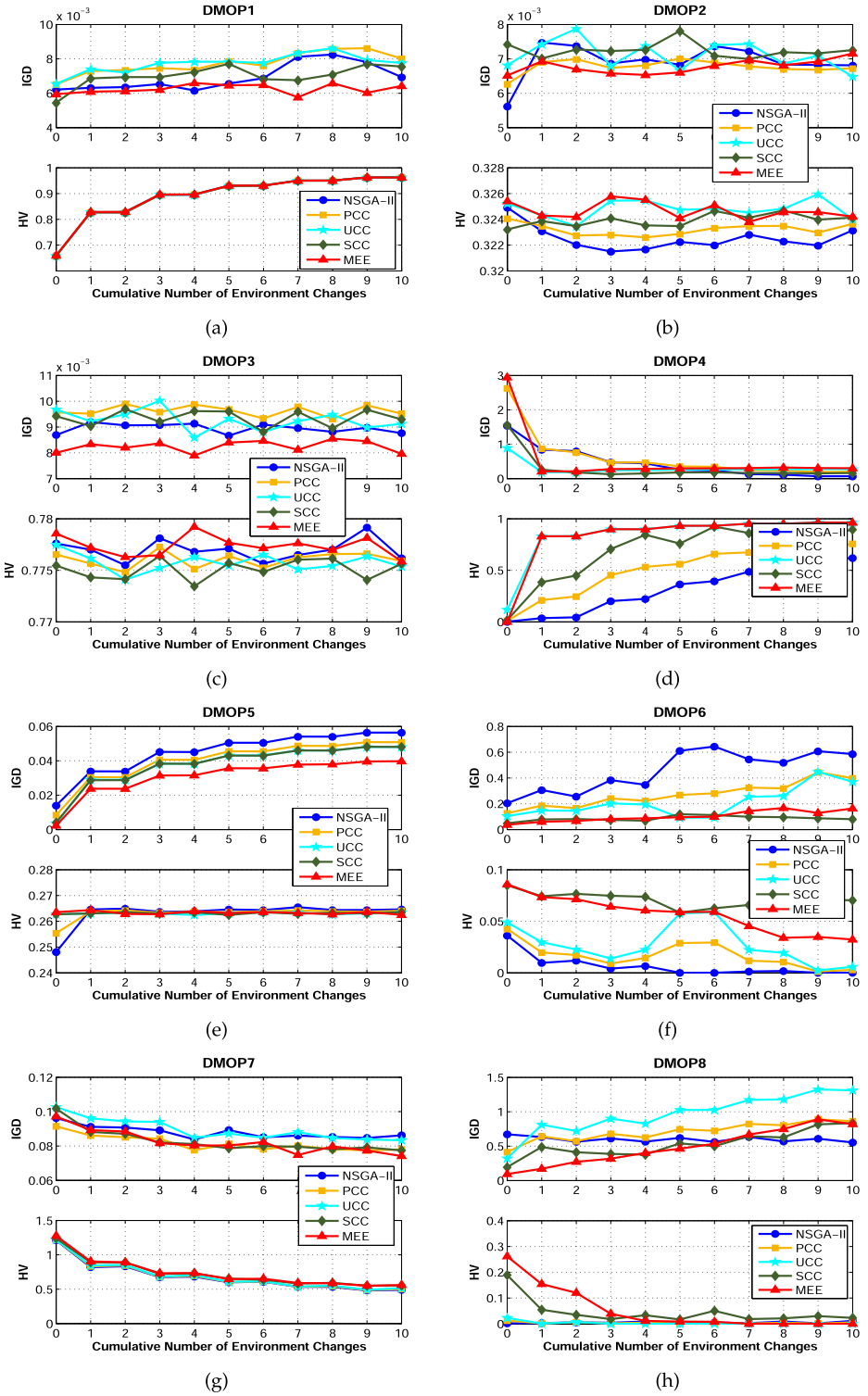

Fig. 3. Averages of IGD and HV over 30 runs versus time on DMOP1-DMOP8.

aspect, are selected to investigate in detail the influence of different strategies for selecting representative individuals on the performance of the proposed method. Table 3 lists the experimental results.

Table 3
Performance comparisons of different strategies of selecting representative solutions with (n_t, τ_t)= (1, 25) and τ =250

Problem	Metric	Random-based strategy	Preference-based strategy	Clustering-based strategy
DMOP1	MSP	0.0131	0.0116	**0.0108**
		(0.0026)†	(0.0028)†	**(0.0022)**
	MGD	0.0035	0.0022	**0.0016**
		(0.0012)†	(0.0013)†	**(0.0009)**
	MMS	0.9706	0.9944	**0.9999**
		(0.0157)†	(0.0148)	**(0.0152)**
DMOP2	MSP	0.0102	0.01	**0.0087**
		(0.0015)†	(0.0021)†	**(0.0013)**
	MGD	0.0088	0.0057	**0.0025**
		(0.0027)†	(0.0016)†	**(0.002)**
	MMS	0.9879	**0.9968**	0.9676
		(0.0165)	**(0.0126)**	(0.0104)†
DMOP6	MSP	0.0242	**0.0195**	0.0207
		(0.0083)†	**(0.0058)**	(0.0067)
	MGD	0.3534	**0.2283**	0.2308
		(0.2522)†	**(0.1704)**	(0.1513)
	MMS	1	1	1
		(0)	(0)	(0)
DMOP7	MSP	0.0408	0.0404	**0.0395**
		(0.0056)	(0.0043)	**(0.005)**
	MGD	0.0258	0.0193	**0.0158**
		(0.0028)†	(0.0032)†	**(0.0038)**
	MMS	1	1	1
		(0)	(0)	(0)

Table 3 reports that, (1) The proposed clustering-based strategy obtains the best values of MSP and MGD in three of four benchmark problems. For DMOP6, the minimal values of MSP and MGD obtained by the preference-based strategy are not significantly different from those obtained by the proposed strategy. (2) All the three strategies have the best values of MMS, 1, for DMOP6 and DMOP7. For DMOP1, the proposed strategy significantly outperforms the random-based one in terms of MMS. For DMOP2, the proposed strategy is slightly worse than the other two in terms of MMS.

In general, the proposed strategy has the best performance among all the three algorithms. That is, we can draw the following conclusions: the performances of the proposed algorithm in terms of diversity and convergence can be improved by adopting the proposed cluster-based strategy of selecting representative solutions. Therefore, we choose MIGD and MHV as the metrics to evaluate the performance of each algorithm in the subsequent experiments, because they can reflect the performance of an algorithm in convergence and diversity.

4.8. Comparison with MOPSO and SPEA2

In this section, two proposed algorithms, MEE-MOPSO and MEE-SPEA2, are compared with MOPSO and SPEA2. Table 4 shows metrics MIGD and MHV in terms of the mean and standard deviation achieved by different algorithms on the DMOP1-DMOP8.

Table 4
The experimental results in terms of the MIGD and MHV metrics of MOPSO, SPEA2, MEE-MOPSO and MEE-SPEA2

Problem	*MIGD*				*MHV*			
	MOPSO	SPEA2	MEE-MOPSO	MEE-SPEA2	MOPSO	SPEA2	MEE-MOPSO	MEE-SPEA2
DMOP1	0.0074	0.0075	**0.0062**	0.0071	0.8699	0.7831	**0.8800**	0.8702
	(0.0003)†	(0.0002)†	**(0.0001)**	(0.0003)	(0.0003)	(0.0003)†	**(0.0005)**	(0.0003)
DMOP2	0.0071	0.0070	**0.0061**	0.0070	0.3230	0.2925	0.3241	**0.3243**
	(0.0002)†	(0.0002)†	**(0.0001)**	(0.0003)	(0.0005)	(0.0003)	(0.0007)	**(0.0007)**
DMOP3	0.0093	0.0092	**0.0054**	0.0089	0.7754	0.6985	**0.7773**	0.7770
	(0.0002)†	(0.0002)†	**(0.0001)**	(0.0001)†	(0.0006)	(0.0004)†	**(0.0003)**	(0.0006)
DMOP4	0.3489	0.4891	0.4352	**0.2021**	0.2994	0.7212	0.6690	**0.7608**
	(0.0930)†	(0.0501)†	(0.0226)†	**(0.0370)**	(0.0224)†	(0.0218)	(0.0990)†	**(0.0002)**
DMOP5	0.0406	0.0412	**0.0406**	0.0406	0.2631	0.2368	0.2628	**0.2632**
	(0.0001)	(0.0026)	**(0.0001)**	(0,0001)	(0.0003)	(0.0002)	(0.0010)	**(0.0042)**
DMOP6	0.2354	0.3778	0.1203	**0.0791**	0.0323	0.0102	0.0574	**0.0655**
	(0.0265)†	(0.0264)†	(0.0106)†	**(0.0042)**	(0.0039)†	(0.0019)†	(0.0035)	**(0.0014)**
DMOP7	0.0882	0.0878	**0.0827**	0.0839	0.7487	0.6981	**0.7806**	0.7287
	(0.0040)	(0.0017)	**(0.0008)**	(0.0012)	(0.0191)	(0.0027)†	**(0.0034)**	(0.0106)
DMOP8	0.9313	0.4524	0.5487	**0.3742**	0.0101	0.0579	**0.0660**	0.0573
	(0.1057)†	(0.0873)†	(0.0560)†	**(0.0221)**	(0.0072)†	(0.0107)	**(0.0135)**	(0.0193)

Table 4 reports that, with the help of the proposed grouping strategy, MEE-MOPSO and MEE-SPEA2 perform well on DMOP1-DMOP8 and are followed by MOPSO and SPEA2. On DMOP6, the MIGD values of MEE-SPEA2 and MEE-MOPSO are 0.07908 and 0.12031, respectively. They are all smaller than 0.23535 of MOPSO and 0.37779 of SPEA2. Therefore, MEE-MOPSO and MEE-SPEA2 have a better capability in distribution and convergence than the other two algorithms. Although there is no significantly different in terms of MIGD and MHV metrics among the four algorithms on DMOP5 and DMOP7, MEE-MOPSO and MEE-SPEA2 gain the lowest MIGD values. The reason may lie in that the presented response strategy is not very sensitive to the situation that the true Pareto-optimal sets of DMOP5 and DMOP7 change are not influenced by the number of changing decision variables.

4.9. Comparison with MOEA/DVA and LMEA

In this subsection, the presented algorithm, MEE-NSGA-II, is compared with MOEA/DVA [22] and LMEA [21] on the eight benchmark problems, DMOP1-DMOP8, in which the maximum number of decision variables is set to 1010 ($S = 1000, s = 10$), and the initial number of decision variable, D, is 1005, respectively. During the experiments, the maximum number of function evaluations is set to 20,000,000 for problems with 1000 decision variables when the environment changes. Table 5 shows the values of MIGD and MHV in terms of the mean and standard deviation achieved by the comparison algorithms on the eight problems. As reported in Table 5, MEE-NSGA-II is significantly better than MOEA/DVA on all the eight problems but DMOP4 in terms of MIGD and MHV. Taking DMOP6 as an example, the MIGD value of MEE-NSGA-II is 0.1881, which is clearly smaller than that of MOEA/DVA, 0.4964. Although MOEA/DVA obtains the maximum value of MHV on DMOP1, there is no significant difference between MOEA/DVA (0.4799) and MEE-NSGA-II (0.4798). Therefore, MEE-NSGA-II has better performance in convergence and diversity than MOEA/DVA on DMOP1-DMOP8 except DMOP4. With respect to the comparison between LMEA and MEE-NSGA-II, LMEA is significantly superior to MEE-NSGA-II on DMOP5 and DMOP6 in terms of MIGD. However, it is significantly inferior to MEE-NSGA-II on DMOP1, DMOP2 and DMOP3, and they have the same performance in convergence and distribution on DMOP7 and DMOP8. MEE-NSGA-II is slightly better than LMEA on DMOP4. Therefore, MEE-NSGA-II has better performance than LMEA. In summary, we focus mainly on DMOPs in this paper, suggesting that rapidly tracking the time-dependent PF of an optimization problem is of necessity, which requires that the obtained solutions to a DMOP may be not optimal. The existing methods, such as MOEA/DVA and LMEA, consume plenty of function evaluations when re-investigating the correlation between decision variables, and further adjust groups when the optimization problem changes, suggesting that they have difficulties in meeting the requirements of DMOPs. In contrast, the proposed algorithm saves a large number of function evaluations, leading to more evaluations for improving the performance in convergence and distribution. In addition, information provided by sub-populations has been fully taken advantage of and the initial groups are just locally adjusted as the optimization problem changes. From the above viewpoints, the proposed grouping method is more suitable for MOPs with changing decision variables.

5. Application in a Multi-period Portfolio Selection Problem

In this section, we consider a multi-period portfolio selection problem in emerging markets [48]. To help investors to make competent decisions, we formulate the problem with dynamics as a bi-objective optimization model with changing decision variables. In the formulated model, the expected return rate and risk loss rate at the tth period are represented as follows.

Table 5
The values of MIGD and MHV obtained by MOEA/DVA, LMEA and MEE-NSGA-II.

Problem	*MIGD*			*MHV*		
	MOEA/DVA	LMEA	MEE-NSGA-II	MOEA/DVA	LMEA	MEE-NSGA-II
DMOP1	0.0408	0.0414	**0.0394**	**0.4799**	0.4797	0.4798
	(0.0002)†	(0.0002)†	**(0.0002)**	**(0.0018)**	(0.0012)	(0.0013)
DMOP2	0.0406	0.0407	**0.0393**	0.2158	0.2145	**0.2165**
	(0.0003)	(0.0002)	**(0.0001)**	(0.0004)†	(0.0006)†	**(0.0000)**
DMOP3	0.0452	0.0453	**0.0426**	0.4329	0.4338	**0.4432**
	(0.0004)†	(0.0002)†	**(0.0001)**	(0.0002)†	(0.0001)†	**(0.0001)**
DMOP4	**0.0426**	0.0455	0.0444	**0.1804**	0.1773	0.1796
	(0.0001)	(0.0001)†	(0.0003)†	**(0.0001)**	(0.0006)†	(0.0004)†
DMOP5	0.8173	**0.2686**	0.3102	0.0156	0.0360	**0.0394**
	(0.0105)†	**(0.0040)**	(0.0098)†	(0.0009)†	(0.0003)†	**(0.0004)**
DMOP6	0.4964	**0.1232**	0.1881	0.0235	**0.0478**	0.0395
	(0.0226)†	**(0.0049)**	(0.0041)†	(0.0018)†	**(0.0007)**	(0.0011)†
DMOP7	0.1336	0.1271	**0.1244**	0.4898	0.5154	**0.5159**
	(0.0017)†	(0.0017)	**(0.0006)**	(0.0097)†	(0.0011)	**(0.0018)**
DMOP8	0.1798	**0.1703**	0.1710	0.4015	**0.4289**	0.4266
	(0.0012)†	**(0.0012)**	(0.0020)	(0.0026)†	**(0.0016)**	(0.0017)

Table 6
Comparisons of different algorithms on the portfolio selection problem.

Metric	NSGA-II	SPEA2	MOEA/DVA	LMEA	MEE-NSGA-II
MHV	0.0073	0.0072	0.0070	0.0072	**0.0077**
(std.)	(0.00020)†	(0.00017)†	(0.00017)†	(0.00018)†	(**0.00013**)

$R(x(t), r(t)) = \sum_{i=1}^{n} r_{t,i}x_{t,i} - \sum_{i=1}^{n} a_{t,i}|x_{t,i} - x_{t-1,i}| + r_{t,0}x_{t,0}$, $Q(x(t), q(t)) = \sum_{i=1}^{n} q_{t,i}x_{t,i}$.
Therefore, the bi-objective optimization model can be formulated as

$$\begin{aligned} &\min(-R(x(t), r(t)), Q(x(t), q(t))) \\ &s.t. \sum_{i=1}^{n} x_{i,t} = 1, x_{i,t} \geqslant 0, \quad t = 1, 2, \ldots, T. \end{aligned} \tag{9}$$

where $r(t), q(t), a_{t,i}$ are parameters, whose meaning and setting can be found in [48], and $x_{t,i}$ refers to the investment proportion of the ith security at the tth period.

The proposed algorithm, MEE-NSGA-II, and the comparative ones are employed to handle the multi-period portfolio selection problem. When solving the above problem, the number of securities randomly increases or decreases at each new period. Each algorithm is run 20 times independently. We record these results and calculate their mean. The reference point is set to (0, 0.2) when computing hyper-volume. The MHV value of each comparative algorithm is listed in Table 6. In addition, we show the average HV value obtained by NSGA-II, SPEA2, MOEA/DVA, LMEA, and MEE-NSGA-II over 20 runs versus time, in Fig. 4, to intuitively demonstrate the advantages of the proposed algorithm.

Table 6 indicates that MEE-NSGA-II achieves the best MHV value among these algorithms. In addition, there is a significant difference between MEE-NSGA-II and NSGA-II, SPEA2, MOEA/DVA, and LMEA. Moreover, MEE-NSGA-II has a smaller standard deviation than NSGA-II, MOEA/DVA, and LMEA.

From Fig. 4, we can obtain that although the proposed algorithm, MEE-NSGA-II, does not achieve the best HV value at t = 0, there is a very slight difference between MEE-NSGA-II and the comparative algorithms. Furthermore, MEE-NSGA-II achieved the best HV values among all algorithms from $t = 1$ to $t = 10$, indicating that MEE-NSGA-II has the best performance in terms of convergence and distribution. From the above analysis, we can conclude that the proposed algorithm can achieve a Pareto optimal set with good performance in convergence and diversity. Furthermore, it exhibits stable performance in rapidly tracking time-variant Pareto fronts.

6. Conclusion

To solve a DMOP with varying decision variables, we presented a CCEA, termed MEE-NSGA-II, based on dynamically classifying variables according to the MEE. In MEE-NSGA-II, a complex DMOP is disintegrated into several relatively simple subproblems by classifying variables, resulting in reduced computational complexity. In contrast to other decomposition methods, the proposed MEE-based method guarantees that variables with high dependency are assigned to the same group, with the purpose of efficiently evolving each sub-population. In addition, a method of responding to varying decision variables is proposed for MEE-NSGA-II, which uses different numbers of solutions in the archive to initialize sub-populations corresponding to different groups, and a clustering-based method of selecting representative solutions. These two strategies enhance the proposed algorithm for tracking moving PF.

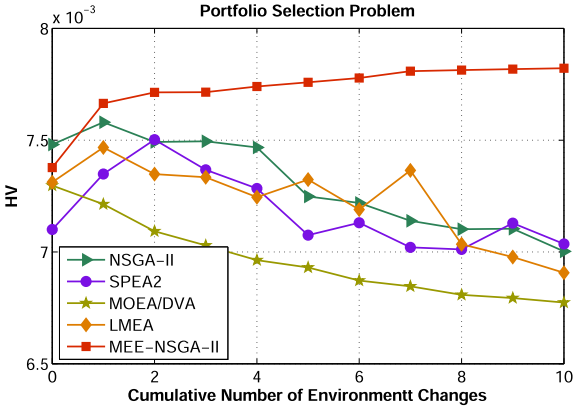

Fig. 4. Average of HV over 20 runs versus time on the portfolio selection problem.

To investigate the performance of MEE-NSGA-II, we applied it to tackle eight benchmark optimization problems, DMOP1-DMOP8, in comparison with five popular algorithms. The experimental results show that MEE-NSGA-II is highly competitive among the comparative algorithms for most problems. Furthermore, we also applied the proposed algorithm in a multi-period portfolio selection problem, and obtained stable performance in rapidly tracking time-variant Pareto fronts.

It is worth mentioning that we only consider an optimization problem with increasing or decreasing one decision variable at a time in this study. However, there may be dynamic optimization problems with other characteristics, such as the deletion of more than one decision variable and simultaneous addition of other variables. In addition, the frequency of the change and the additional/reduced number of decision variables in the above two scenarios may be irregular. To address these problems, new efficient methods are required, which will be the focus of our future work.

CRediT authorship contribution statement

Biao Xu: Supervision, Project administration. **Yong Zhang:** Software, Writing – original draft. **Shengxiang Yang:** Data curation, Investigation. **Ling Wang:** Validation. **Zhun Fan:** Funding acquisition. **Yonggang Zhang:** Writing – review & editing. **Dunwei Gong:** Conceptualization, Methodology.

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.

Acknowledgements

This work is jointly supported by National Key R&D Program of China (2021YFE0199000); National Natural Science Foundation of China (61973305, 62133015, 62176147); State Key Lab of Digital Manufacturing Equipment and Technology (DMETKF2019020); Project of Robot Automatic Design Platform combining Multi-Objective1 Evolutionary Computation and Deep Neural Network (2019A050519008); Natural Science Foundation of Guangdong Province (2021A1515011709); Fundamental Research Funds for the Central Universities, JLU (93K172021K13); Scientific Research Staring Foundation of Shantou University (NTF20009, NTF21001); Natural Science Foundation of the Anhui Higher Education Institutions of China (KJ2019A0956); Major Project of Natural Science Research of the Jiangsu Higher Education Institutions of China (21KJA520006); and Xuzhou Science and Technology Plan Project (KC21007).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.ins.2022.05.123.

References

[1] S. Jiang, H. Li, J. Guo, M. Zhong, S. Yang, M. Kaiser, N. Krasnogor, Area: An adaptive reference-set based evolutionary algorithm for multiobjective optimisation, Inf. Sci. 515 (2020) 365–387.
[2] D. Gong, B. Xu, Y. Zhang, Y. Guo, S. Yang, A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems, IEEE Trans. Evol. Comput. 24 (1) (2020) 142–156.
[3] M.K. Mehlawat, P. Gupta, A.Z. Khan, Portfolio optimization using higher moments in an uncertain random environment, Inf. Sci. 567 (2021) 348–374.
[4] Q. Wu, X. Liu, J. Qin, L. Zhou, Multi-criteria group decision-making for portfolio allocation with consensus reaching process under interval type-2 fuzzy environment, Inf. Sci. 570 (2021) 668–688.
[5] Y. Hu, J. Zheng, J. Zou, S. Jiang, S. Yang, Dynamic multi-objective optimization algorithm based decomposition and preference, Inf. Sci. 571 (2021) 175–190.
[6] K. Deb, B.R.N. Udaya, S. Karthik, Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling, in: Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 803–817.
[7] Z. Liang, S. Zheng, Z. Zhua, S. Yang, Hybrid of memory and prediction strategies for dynamic multiobjective optimization, Inf. Sci. 485 (2019) 200–218.
[8] F. Zou, G.G. Yen, C. Zhao, Dynamic multiobjective optimization driven by inverse reinforcement learning, Inf. Sci. 575 (2021) 468–484.
[9] M. Rong, D. Gong, Y. Zhang, Y.and Jin, W. Pedrycz, Multi-directional prediction approach for dynamic multi-objective optimization problems, IEEE Transactions on Cybernetics 49 (9) (2019) 3362–3374.
[10] R. Chen, K. Li, X. Yao, Dynamic multi-objectives optimization with a changing number of objectives, IEEE Trans. Evol. Comput. 22 (1) (2018) 157–171.
[11] M. Olhofer, Y. Jin, B. Sendhoff, Adaptive encoding for aerodynamic shape optimization using evolution strategies, in: IEEE Congress on Evolutionary Computation, Vol. 1, IEEE, 2001, pp. 576–583.
[12] Y. Jin, M. Olhofer, B. Sendhoff, On Evolutionary Optimization of Large Problems Using Small Populations, Springer, Berlin Heidelberg, 2005.
[13] M.A. Potter, K.A.D. Jong, A cooperative coevolutionary approach to function optimization, Lect. Notes Comput. Sci. 866 (1994) 249–257.
[14] P. Yang, K. Tang, X. Yao, A parallel divide-and-conquer-based evolutionary algorithm for large-scale optimization, IEEE Access 7 (2019) 163105–163118.
[15] M.N. Omidvar, X. Li, Y. Mei, X. Yao, Cooperative co-evolution with differential grouping for large scale optimization, IEEE Trans. Evol. Comput. 18 (3) (2014) 378–393.
[16] Y.-H. Jia, W.-N. Chen, T. Gu, H. Zhang, H.-Q. Yuan, S. Kwong, J. Zhang, Distributed cooperative co-evolution with adaptive computing resource allocation for large scale optimization, IEEE Trans. Evol. Comput. 23 (2) (2019) 188–202.
[17] K. Li, R. Chen, G. Min, X. Yao, Integration of preferences in decomposition multiobjective optimization, IEEE Trans. Cybern. 48 (12) (2018) 3359–3370.

[18] K. Li, H. Nie, H. Gao, X. Yao, Posterior decision-making based on decomposition-driven knee point identification, IEEE Trans. Evol. Comput. (2021), https://doi.org/10.1109/TEVC.2021.3116121.
[19] K.C. Tan, Y.J. Yang, T.H. Lee, A distributed cooperative coevolutionary algorithm for multiobjective optimization, in, IEEE Congress on Evolutionary Computation, IEEE (2004) 2513–2520.
[20] C.-K. Goh, K.C. Tan, A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, IEEE Trans. Evol. Comput. 13 (1) (2009) 103–127.
[21] X. Zhang, Y. Tian, R. Cheng, Y. Jin, A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization, IEEE Trans. Evol. Comput. 22 (1) (2018) 97–112.
[22] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, M. Gong, A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables, IEEE Trans. Evol. Comput. 20 (2) (2016) 275–298.
[23] M.N. Omidvar, X. Li, Y. Mei, X. Yao, Cooperative co-evolution with differential grouping for large scale optimization, IEEE Trans. Evol. Comput. 18 (3) (2014) 378–393.
[24] M.N. Omidvar, M. Yang, Y. Mei, X. Li, X. Yao, DG2: A faster and more accurate differential grouping for large-scale black-box optimization, IEEE Trans. Evol. Comput. 21 (6) (2017) 929–942.
[25] Y. Mei, M.N. Omidvar, X. Li, X. Yao, A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization, ACM Transactions on Mathematical Software 42 (2) (2016) 13:1–13:24..
[26] Y. Sun, M. Kirley, S.K. Halgamuge, A recursive decomposition method for large scale continuous optimization, IEEE Trans. Evol. Comput. 22 (5) (2018) 647–661.
[27] Y. Sun, X. Li, A. Ernst, M.N. Omidvar, Decomposition for large-scale optimization problems with overlapping components, in, IEEE Congress on Evolutionary Computation (CEC) 2019 (2019) 326–333.
[28] Y. Sun, M.N. Omidvar, M. Kirley, X. Li, Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition, in, Genetic and Evolutionary Computation Conference (GECCO) (2018) 889–896.
[29] B. Xu, Y. Zhang, D. Gong, L. Wang, A parallel multi-objective cooperative co-evolutionary algorithm with changing variables, in, Genetic and Evolutionary Computation Conference Companion, ACM (2017) 1888–1893.
[30] T.M. Cover, J.A. Thomas, Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing), Wiley-Interscience, 2017.
[31] D. Reshef, H.K. Reshef, A. Yakir, Finucane: Detecting novel associations in large data sets, Science 334 (6062) (2011) 1518–1524.
[32] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[33] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, TIK-report 103 (2001).
[34] Y. Zhang, D. Gong, J. Cheng, Multi-objective particle swarm optimization approach for cost-based feature selection in classification, IEEE/ACM Trans. Comput. Biol. Bioinf. 14 (1) (2017) 64–75.
[35] X. Zhang, X. Zheng, R. Cheng, J. Qiu, Y. Jin, A competitive mechanism based multi-objective particle swarm optimizer with fast convergence, Inf. Sci. 427 (2018) 63–76.
[36] C.K. Goh, K.C. Tan, D. Liu, S.C. Chiam, A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design, Eur. J. Oper. Res. 202 (1) (2010) 42–54.
[37] F. Caraffini, F. Neri, L. Picinali, An analysis on separability for memetic computing automatic design, Inf. Sci. 265 (2014) 1–22.
[38] Y. Sun, M. Kirley, S.K. Halgamuge, Quantifying variable interactions in continuous optimization problems, IEEE Trans. Evol. Comput. 21 (2) (2017) 249–264.
[39] F. Wang, F. Liao, Y. Li, H. Wang, A new prediction strategy for dynamic multi-objective optimization using Gaussian mixture model, Inf. Sci. 580 (2021) 331–351.
[40] B. Xu, Y. Zhang, D. Gong, Y. Guo, M. Rong, Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization, IEEE/ACM Trans. Comput. Biol. Bioinf. 15 (6) (2018) 1877–1890.
[41] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 256–279.
[42] K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, Evol. Comput. 7 (3) (1999) 205–230.
[43] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[44] L.C. Jiao, H. Wang, R.H. Shang, F. Liu, A co-evolutionary multi-objective optimization algorithm based on direction vectors, Inf. Sci. 228 (7) (2013) 90–112.
[45] Y. Tian, T. Zhang, J. Xiao, X. Zhang, Y. Jin, A coevolutionary framework for constrained multiobjective optimization problems, IEEE Trans. Evol. Comput. 25 (1) (2021) 102–116.
[46] H. Ma, H. Wei, Y. Tian, R. Cheng, X. Zhang, A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints, Inf. Sci. 560 (2021) 68–91.
[47] Y. Tian, X. Zhang, C. Wang, Y. Jin, An evolutionary algorithm for large-scale sparse multiobjective optimization problems, IEEE Trans. Evol. Comput. 24 (2) (2020) 380–393.
[48] J. Sun, X. Yan, Z. Heng, L. Zhiping, Interval multi-objective programming methods for solving multi-period portfolio selection problems, Control Decis. 35 (3) (2020) 645–650 (in Chinese).