A Similarity-based Cooperative Co-evolutionary Algorithm for Dynamic Interval Multi-objective Optimization Problems

Dunwei Gong, *Member, IEEE,* Biao Xu, Yong Zhang, Yinan Guo, and Shengxiang Yang, *Senior Member, IEEE*

Abstract—Dynamic interval multi-objective optimization problems (DI-MOPs) are very common in real-world applications. However, there are few evolutionary algorithms that are suitable for tackling DI-MOPs up to date. A framework of dynamic interval multi-objective cooperative co-evolutionary optimization based on the interval similarity is presented in this paper to handle DI-MOPs. In the framework, a strategy for decomposing decision variables is first proposed, through which all the decision variables are divided into two groups according to the interval similarity between each decision variable and interval parameters. Following that, two sub-populations are utilized to cooperatively optimize decision variables in the two groups. Furthermore, two response strategies, i.e., a strategy based on the change intensity and a random mutation strategy, are employed to rapidly track the changing Pareto front of the optimization problem. The proposed algorithm is applied to eight benchmark optimization instances as well as a multi-period portfolio selection problem and compared with five state-of-the-art evolutionary algorithms. The experimental results reveal that the proposed algorithm is very competitive on most optimization instances.

Index Terms—Multi-objective optimization, dynamic optimization, cooperative co-evolutionary optimization, interval similarity, response strategy.

I. INTRODUCTION

THERE are various multi-objective optimization problems (MOPs) with the interval characteristic in real-word applications. Each of these optimization problems generally contains more than one objective conflicting with each other, and has the interval characteristic in at least one objective and (or) constraint. One representative instance is production planning in a steel-making continuous casting-hot rolling (SCC-HR) process [1]. The process can be formulated as an MOP with the interval characteristic. For this problem, there are various uncertainties in the production process, e.g., the processing time, the production leading time, to name a few,

which are contained in such objectives as the throughput, the hot charge ratio, the utilization rate, and the additional cost, and embodied with intervals. Another typical instance is robot path planning. When planning the path of a robot, the workspace of the robot often involves various danger sources, such as fire, landmines and enemies. Given the fact that it is too expensive or even impossible to get the precise positions of these danger sources, decision-makers only know their ranges in most cases. Along this line, taking two performance metrics, i.e. the risk and the path distance, into consideration, the problem of robot path planning in the environment with uncertain danger sources can be formulated as a bi-objective optimization problem with interval parameters [2]. To illustrate the popularity of MOPs with the interval characteristic, let us investigate the reliability redundancy allocation problem. For this problem, the reliability of an individual component may be imprecise, and is generally represented with an interval, which results in an interval MOP where the reliability of the whole system and its cost are simultaneously optimized [3].

In practice, an uncertain optimization problem can also be modeled as stochastic programming [4], [5], [6] or fuzzy programming [7], [8], [9], [10] instead of interval programming. However, additional functions or information are required when formulating the problem, such as probability distribution in stochastic programming and membership functions in fuzzy programming. On the one hand, additional functions or information generally result in a complex model. On the other hand, much history information is required for describing probability distribution and membership functions, which is often hard to obtain. Furthermore, probability distribution and membership functions can be converted to intervals using the confidence level and cut set, respectively. On this circumstance, stochastic and fuzzy optimization problems can be transformed into interval optimization problems.

Interval programming is employed to tackle problems with the interval characteristic. These optimization problems generally contain a number of parameters represented with intervals [11], [12], [13], [14], and the left and right endpoints or the midpoints and radius of these intervals are known a priori. It is relatively easier to obtain an interval than a probability distribution or membership function. When tackling an interval MOP, previous studies generally convert it to a deterministic single- or multi-objective optimization problem [11], [15], [16], [17]. The converted problem, however, is largely different from the original one. In addition, for the same interval MOP, different transformation approaches generally bring about dif-

D. Gong, B. Xu, Y. Zhang, and Y. Guo are with School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221116, China (e-mail: dwgong@vip.163.com; xubiao512@163.com; yongzh401@126.com; guoyinan@cumt.edu.cn). S. Yang is with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, U.K. (e-mail: syang@dmu.ac.uk).
D. Gong is also with School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, 266061, China. And B. Xu is also with School of Mathematical Sciences, Huaibei Normal University, Huaibei, 235000, China.

ferent deterministic optimization problems. As a consequence, optimal solutions to these deterministic problems may be too diverse, which raises a big problem when selecting optimal solutions from a number of solution sets. Compared with the conversion approach, the direct approach can avoid losing valuable information and adding redundant information, thus obtaining more precise solutions.

An MOP with changing ranges of interval parameters in at least one objective and (or) constraint over time is called as a dynamic interval multi-objective optimization problem (DI-MOP). Taking the problem of robot path planning as an example, the ranges of interval parameters will change when the positions of danger sources change over time. At this point, decision-makers should rapidly adjust the robot path so as to successfully fulfill a mission.

Different from traditional dynamic MOPs (DMOPs) with precise objectives, a DI-MOP generally has objectives with interval values, which makes previous approaches, e.g., selecting non-dominated solutions, detecting whether an optimization problem changes or not, and responding the change, unsuitable for this problem. In addition, an algorithm is required to have the capability of obtaining a set of solutions with good performance in convergence and distribution when the problem changes. As a result, it is meaningful and greatly needed to develop efficient approaches to solving DI-MOPs.

Various studies have shown that the co-evolutionary mechanism is beneficial to increasing the efficiency of an optimization process [18], [19]. A cooperative co-evolutionary algorithm (CCEA) generally decomposes an optimization problem with a large number of decision variables into a number of sub-problems with a small number of decision variables, with each being optimized by a sub-population. Each sub-population seeks the optimal solutions for a sub-problem in the corresponding search space. A complete solution, i.e., the candidate to the original optimization problem, for a sub-population is constructed by combining a best solution of the current sub-population with the best solutions of other sub-populations at each generation. Since CCEAs can significantly reduce the search space of a sub-population which is utilized to optimize a small number of decision variables, they are efficient when solving traditional DMOPs. For example, Goh and Tan presented a dynamic competitive-cooperative co-evolutionary algorithm (dCOEA) to solve both static and dynamic MOPs, where all the decision variables are adaptively divided into a number of groups, and stochastic competitors are employed to track changing optimal solutions [20]. Two approaches to large-scale multi-objective optimization were proposed in [21] and [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. But a large number of function evaluations are consumed before the optimization, especially for an optimization problem with a large number of decision variables. It is not suitable for a dynamic problem, which requires an algorithm with the capability in rapidly responding to environmental changes. The proposed decomposition strategy, however, does not take the influence of changing parameters on the decision variables into consideration. Following the influence of the time scale on the decision variables, we divided all the decision variables into two groups [23], among which one contains decision variables interrelated with the time scale, and the other does not. When cooperatively optimizing the decision variables of the two groups using two sub-populations, we employed different prediction strategies to generate the initial population for different sub-populations, with the purpose of rapidly responding to the change of the optimization problem. The above methods focus mainly on real value problems and do not take interval objectives into account, hence incapable to tackle DI-MOPs. Further analyses can be found in Subsections III.C and III.D.

In addition, a non-dominated solution to an interval MOP may not be non-dominated in the scenario of a DI-MOP due to the changing ranges of interval parameters. Consequently, issues are expected to address when solving a DI-MOP, including accurately detecting the change of the optimization problem, rapidly tracking the changing Pareto front (PF), and timely providing candidates with good performance in diversity and approximation.

In this study, we focus on DI-MOPs, and a framework of dynamic interval multi-objective cooperative co-evolutionary optimization based on the interval similarity is presented to handle them. In the framework, a strategy for decomposing decision variables is first proposed, through which all the decision variables are divided into two groups according to the interval similarity between each decision variable and interval parameters. Following that, two sub-populations are utilized to cooperatively optimize decision variables in the two groups. Furthermore, two response strategies, i.e., a strategy based on the change intensity and a random mutation strategy, are employed to rapidly track the changing Pareto front of the optimization problem.

More specifically, this paper has the following four-fold contributions:

(1) Providing a cooperative co-evolutionary optimization framework for tackling DI-MOPs;

(2) Presenting strategies for grouping decision variables of a DI-MOP and detecting its change according to the interval similarity;

(3) Defining the change intensity of an optimization problem and proposing a response strategy based on it, and

(4) Experimentally investigating the performance of the proposed algorithm based on a set of benchmark problems and applying the proposed algorithm to a multi-period portfolio selection problem.

The rest of this paper is structured as follows. Section II provides a comprehensive review on the related work. The proposed framework of dynamic interval multi-objective cooperative co-evolutionary optimization based on the interval similarity is detailed in Section III. The experimental setting and results are reported and analyzed in Section IV. Finally,

Section V concludes the whole paper and points out several topics for future study.

II. THE RELATED WORK

A. DI-MOPs

Without loss of generality, a DI-MOP can be formulated as follows.

$$\min F(\mathbf{x}, \mathbf{c}(t)) = (f_1(\mathbf{x}, \mathbf{c_1}(t)), f_2(\mathbf{x}, \mathbf{c}_2(t)), \cdots, f_m(\mathbf{x}, \mathbf{c}_m(t)))$$
$$\text{s.t. } \mathbf{x} \in \mathbf{D} \subseteq \mathbf{R}^n$$
$$\mathbf{c}_i(t) = (c_{i1}(t), c_{i2}(t), \cdots, c_{il}(t))^{\mathrm{T}}, \ i = 1, 2, \cdots, m$$
$$c_{ik}(t) = [\underline{c_{ik}}(t), \overline{c_{ik}}(t)], k = 1, 2, \cdots, l \tag{1}$$

where

$\mathbf{x}$	- an n-dimensional decision vector;
$\mathbf{c}(t)$	- an m-dimensional interval parameter vector;
$F(\mathbf{x}, \mathbf{c}(t))$	- an objective vector;
$f_i(\mathbf{x}, \mathbf{c}_i(t))$	- the i-th component of $F(\mathbf{x}, \mathbf{c}(t))$;
$\mathbf{c}_i(t)$	- the i-th component of $\mathbf{c}(t)$, which is a vector with l components;
$c_{ik}(t)$	- the k-th component of $\mathbf{c}_i(t)$, which is an interval;
$\underline{c_{ik}}(t)$	- the left endpoint of $c_{ik}(t)$;
$\overline{c_{ik}}(t)$	- the right endpoint of $c_{ik}(t)$.

Given the fact that $\mathbf{c}_i(t)$ is an interval vector, $f_i(\mathbf{x}, \mathbf{c}_i(t))$ is an objective with its value being within an interval, denoted as $f_i(\mathbf{x}, \mathbf{c}_i(t)) \stackrel{\Delta}{=} [\underline{f_i}(\mathbf{x}, \mathbf{c}_i(t)), \overline{f_i}(\mathbf{x}, \mathbf{c}_i(t))], \ i = 1, 2, \cdots, m$. $\mathbf{c}_i(t)$ generally changes over the time scale. Specially, if $\mathbf{c}_i(t)$ remains unchanged over time for any i, Eq. (1) will be degraded as a traditional interval MOP [24]. In addition, if $\underline{c_{ik}}(t) = \overline{c_{ik}}(t)$ is held for any i and k, Eq. (1) will be degraded as a traditional DMOP [25]. From this viewpoint, the optimization problem formulated with Eq. (1) is an extension of previous optimization problems. As a result, studying approaches suitable for Eq. (1) is of considerable significance.

B. Dominance relation and Crowding distance based on intervals

For two solutions, $\mathbf{x}_1$, $\mathbf{x}_2 \in D$, of (1), their i-th objectives are $f_i(\mathbf{x}_1, \mathbf{c}_i(t))$ and $f_i(\mathbf{x}_2, \mathbf{c}_i(t)), i = 1, 2, \cdots, m$, respectively. Limbourg and Aponte [24] defined the following order and dominance relations based on intervals.

Definition 1: Order relation. The order relation between $f_i(\mathbf{x}_1, \mathbf{c}_i(t))$ and $f_i(\mathbf{x}_2, \mathbf{c}_i(t))$ is defined as follows.

$$f_i(\mathbf{x}_1, \mathbf{c}_i(t)) <_{IN} f_i(\mathbf{x}_2, \mathbf{c}_i(t)) \Leftrightarrow \underline{f_i}(\mathbf{x}_1, \mathbf{c}_i(t)) \leq \underline{f_i}(\mathbf{x}_2, \mathbf{c}_i(t))$$
$$\wedge \overline{f_i}(\mathbf{x}_1, \mathbf{c}_i(t)) \leq \overline{f_i}(\mathbf{x}_2, \mathbf{c}_i(t)) \wedge f_i(\mathbf{x}_1, \mathbf{c}_i(t)) \neq f_i(\mathbf{x}_2, \mathbf{c}_i(t)) \tag{2}$$

Otherwise, $f_i(\mathbf{x}_1, \mathbf{c}_i(t))$ and $f_i(\mathbf{x}_2, \mathbf{c}_i(t))$ are incomparable, denoted as $f_i(\mathbf{x}_1, \mathbf{c}_i(t)) || f_i(\mathbf{x}_2, \mathbf{c}_i(t))$.

The order relation, $<_{IN}$, is antisymmetric, reflexive, and transitive. As a result, it defines a partial order relation between intervals.

Definition 2: Dominance relation based on intervals. $\mathbf{x}_1$ is said to dominate $\mathbf{x}_2$ based on intervals, denoted as $\mathbf{x}_1 \succ_{IP} \mathbf{x}_2$, if and only if

$$\forall i \in \{1, 2, \cdots, m\},$$
$$f_i(\mathbf{x}_1, \mathbf{c}_i(t)) <_{IN} f_i(\mathbf{x}_2, \mathbf{c}_i(t)) \vee f_i(\mathbf{x}_1, \mathbf{c}_i(t)) || f_i(\mathbf{x}_2, \mathbf{c}_i(t)),$$
$$\exists i \in \{1, 2, \cdots, m\}, f_i(\mathbf{x}_1, \mathbf{c}_i(t)) <_{IN} f_i(\mathbf{x}_2, \mathbf{c}_i(t)) \tag{3}$$

If neither $\mathbf{x}_1$ dominates $\mathbf{x}_2$, nor $\mathbf{x}_1$ is dominated by $\mathbf{x}_2$, $\mathbf{x}_1$ and $\mathbf{x}_2$ are non-dominated based on intervals, denoted as $\mathbf{x}_1 || \mathbf{x}_2$.

To obtain a diverse Pareto front, Limbourg and Aponte provided the definitions of the hyper-volume and crowding distance based on intervals [24]. Following the dominance relation and the crowding distance based on intervals, they proposed the following strategy for sorting solutions to (1). The dominance relation is first utilized to assign each solution with a unique rank. Then, for solutions with the same rank, the order relation is adopted to sort them according to their crowding distances. Finally, a random strategy is employed to sort solutions which cannot be distinguished by the above approach.

C. Interval multi-objective evolutionary optimization

Various real-world applications can be formulated as interval optimization problems, such as optimal dispatch of a virtual power plant [26], aircraft wing and automobile design [27], [28], and household load scheduling [29]. As a result, seeking approaches suitable for addressing interval optimization problems is of great significance in theory and applications, and among which interval analysis is one of representative tools [30].

In recent years, utilizing evolutionary algorithms (EAs) to address interval MOPs has become a rapidly growing field, and a plenty of achievements have been obtained. Roughly, there are the following two ways to handle interval MOPs with EAs. One is converting an interval MOP to a deterministic single- or multi-objective optimization problem, followed by solving the converted one using EAs. Along this line, Cheng et al. first transformed an interval MOP into a mini-max optimization problem [31]. Then, they adopted a hierarchical algorithm composed of a genetic algorithm and a nonlinear programming approach to tackle the converted problem. Jiang et al. converted an interval MOP to a deterministic MOP based on the middle point and the width of an interval [32], and further converted it to a single-objective optimization without constraints which is tackled by an EA. We converted an interval MOP with hybrid indices to a deterministic MOP using such information as the middle point and the width of an interval [15], and employed NSGA-II [33] to address the converted problem. Sahoo et al. first formulated an MOP with interval parameters, followed by converting the model to a single-objective optimization problem and solving it using an improved EA [17]. In addition, Bhunia et al. first defined the interval ordering relation between solutions to an interval MOP. Then, they transformed the interval MOP into a deterministic single-objective optimization problem and tackled the

converted problem using a hybrid EA [16]. Recently, focusing on an interval many-objective optimization problem, we first transformed it into a deterministic bi-objective problem, where new objectives are hyper-volume and imprecision. And then, a set-based genetic algorithm was proposed to tackle it [11].

The other is tackling interval MOPs directly by EAs based on the interval dominance relation. Keeping this line in mind, Limbourg et al. defined an interval Pareto dominance relation based on the interval order relation when tackling an interval MOP [24]. In addition, they proposed an imprecision-propagating multi-objective evolutionary algorithm (IP-MOEA) to address the interval MOP. We proposed a Pareto dominance relation based on the interval possibility and interval crowding distance [34], [35]. Using them, we selected the optimal solutions to an interval MOP. Sun et al. presented an interval Pareto dominance relation based on the lower limit of a possibility, and employed it to modify NSGA-II to cope with interval MOPs [36]. Zhang et al. evaluated solutions using a probability dominance relation [37]. Goh and Tan calculated the probability with which a solution dominates another, and compared solutions based on the probability [38]. Karshenas et al. presented an α-degree Pareto dominance to discriminate solutions to an interval MOP [39]. In addition, Dou et al. presented the scheme of the interval hesitation dominance to distinguish solutions [40]. However, different dominance relations will derive different optimal solution sets for the same interval programming problem even though the identical algorithm is adopted. It is difficult to choose a favorable solution set for users. Bearing this characteristic in mind, Sun et al. integrated previous interval dominance rules in a framework through investigating the correlations of interval dominance rules, developing a strategy to reducing a rule set and assembling the reduced rules. As a result, users can utilize the proposed ensemble dominance to handle their interval programming models. Moreover, they will free themselves from choosing an appropriate approach to assess solutions [41], [42].

D. Dynamic Multi-Objective Optimization Algorithms

For a DMOP, an ideal optimization algorithm is expected to rapidly seek Pareto-optimal solutions before the problem changes, and a well-designed dynamic multi-objective optimization algorithm generally has good performance in convergence and diversity.

Various studies have focused on speeding up the convergence and maintaining a good diversity of EAs from the following two aspects. One is developing memory-based methods, which save relevant information of the current solutions, and employ it in the subsequent stages. Along this line, Goh et al. selected a number of non-dominated solutions from an archive, and re-evaluated them to obtain information for guiding the subsequent evolution [20]. In the immune clonal algorithm (ICA) for a DMOP [43], Shang et al. saved representative individuals as the initial population when the optimization problem changes, so as to accelerate the convergence. They also proposed an improved decomposition-based memetic algorithm in [44], and employed an archive

to store the currently best solution in each decomposition direction during the search. The helpful information offered by the archive can assist in handling neighbor sub-problems by cooperation. Azzouz et al. proposed an adaptive strategy for managing hybrid populations with memory, local search and random strategies, to effectively tackle DMOPs, which guarantees a rapid convergence and good diversity [45]. Koo et al. proposed a selective memory technique, which selects a partial retrieval based on the diversity in the decision space to maintain effective memories [46]. Recently, Chen et al. focused on DMOPs with a time-variable number of objectives in [47], and proposed a dynamic two-archive EA, denoted as DTAEA, to address them. DTAEA simultaneously maintains two co-evolutionary populations, i.e., CA and DA, during the evolution, and CA and DA will be reconstructed as the environment changes. By a restricted mating selection mechanism, DTAEA takes full advantages of complementary influences of CA and DA, with the purpose of striking the balance between convergence and diversity. Additionally, they utilized a truncation operator to retrieve the most diverse subset of memories.

The other is designing prediction-based approaches, which predict optimal solutions based on historical information when the optimization problem changes. Zhou et al. proposed a method of re-initializing a population based on prediction [48]. The method first predicts the new positions of individuals based on information collected during previous searches, and the current population is then partially or completely replaced by the predicted individuals. This method has, however, deteriorative performance when the PS of an optimization problem changes nonlinearly over time. To improve the prediction accuracy, a method of hybrid diversity maintenance was proposed in [49], which firstly relocates a number of solutions close to the new Pareto front by prediction based on the moving direction of each center. Following that, it employs a gradual search to generate a number of well-distributed solutions in the decision space, so as to compensate for possible inaccuracies. To maintain the diversity of a population, the rest solutions are randomly generated. Moreover, Liu et al. proposed an improved prediction model to overcome this drawback [50]. In the proposed model, two individuals are selected to guide the newly predicted individuals, which is beneficial to preventing them from moving toward wrong directions [48]. In addition, Muruganantham et al. presented a dynamic MOEA by utilizing a Kalman filter (KF) to predict the new PF [51], which uses less information than the strategy proposed by Zhou. Furthermore, Rong et al. proposed a multidirectional prediction (MDP) strategy to enhance the performance of EAs in addressing a DMOP. They construct multiple time series models based on historical information to predict a number of evolutionary directions. Once an environmental change occurs, a part of the population is re-initialized by the prediction model, and the rest will be randomly generated [52].

In addition to the aforementioned methods, Shang et al. combined a co-evolutionary competitive and cooperative operation in the immune clonal algorithm (ICA) for DMOPs to have good performance of solutions in uniformity and diversity [53], [54]. Qu et al. proposed an ensemble of multi-objective

---

Algorithm 1: The pseudo code of the proposed framework

   Input: the number of generations, g;
   Output: the archive, $A(t)$;
1 Dividing decision variables (Algorithm 2);
2 $t \leftarrow 0$, $g \leftarrow 0$ % t is the time scale;
3 Initialize sub-populations;
4 **while** *the termination condition is not met* **do**
5   Change detection (Algorithm 3);
6   **if** *the change occurs* **then**
7 $t \leftarrow t+1$;
8   Response to the problem change;
9   **end**
10   Cooperatively co-evolve sub-populations;
11   Select non-dominated solutions and store them in the archive, $A(t)$;
12 $g \leftarrow g+1$;
13 **end**
14 **return** $A(t)$

---

differential evolution algorithms to cope with dynamic economic emission dispatch (DEED) problems [55], [56], and the experimental results demonstrated that the proposed algorithm is capable in speeding up convergence and effective in handling DEED problems.

III. DYNAMIC INTERVAL MULTI-OBJECTIVE COOPERATIVE CO-EVOLUTIONARY OPTIMIZATION BASED ON INTERVAL SIMILARITY

A. The general framework

DI-MOPs have the following twofold characteristics: one is that each objective of a DI-MOP is an interval, and the other is that interval parameters in the objectives change over time. Aiming at these two characteristics, we propose a framework of dynamic interval multi-objective cooperative co-evolutionary optimization based on the interval similarity to handle DI-MOPs.

In the proposed framework, all the decision variables of an optimization problem are first decomposed according to the strategy proposed in Subsection III.C. Next, sub-populations cooperatively co-evolve with each sub-population evolving the decision variables in one group, and a complete solution is formed as described in Subsection III.F with its objective being the fitness of an individual to be evaluated. Following that, non-dominated solutions are selected, and saved in an archive. In addition, a strategy based on the interval similarity is employed to check whether the optimization problem changes or not, as described in Subsection III.D. If yes, the sub-populations will be re-initialized using the proposed response and perturbation strategies, respectively, as described in Subsection III.E. The above process will be repeated until a termination condition is met. The complete non-dominated solutions in the archive are finally output as the optimal solutions to the DI-MOP. Algorithm 1 provides the pseudo code of the framework.

To fulfill this task, we first define the interval similarity as follows.

B. Interval similarity

In [57], the authors provide a definition of interval similarity and its properties. The definition, however, is useless when an interval degrades as a real value. Therefore, we extend the definition and make it also suitable for real values.

Definition 3: Interval similarity. For two intervals $a = [\underline{a}, \overline{a}]$ and $b = [\underline{b}, \overline{b}]$, their similarity, denoted as $s(a, b)$, is defined as follows.

$$
s(a,b) = \begin{cases} \frac{l_{a \cap b}}{\max\{l_a, l_b\}}, & \max\{l_a, l_b\} \neq 0; \\ 1 - \frac{|b-a|}{\max\{|a|,|b|,\chi(|a|+|b|)\}}, & \\ & \max\{l_a, l_b\} = 0, \ a \neq 0 \ or \ b \neq 0; \\ 1, & \max\{l_a, l_b\} = 0, \ a = b = 0. \end{cases}
$$

(4)

where $a \cap b$ represents the intersection of a and b, and l_a, l_b, and $l_{a \cap b}$ mean the widths of intervals a, b, and $a \cap b$, respectively. $\max\{l_a, l_b\} = 0$ indicates that a and b are two real values, i.e. $\underline{a} = \overline{a}$ and $\underline{b} = \overline{b}$. χ is a characteristic function, with its value being 1 when $ab < 0$, and 0 otherwise.

The value of $s(a, b)$ reflects the difference between a and b, which satisfies that $0 \leq s(a, b) \leq 1$, and a larger value means a smaller difference between these intervals.

In addition, we have the following observations.

(1) $s(a, b)=1$ if and only if $a = b$, which is called complete.
(2) $s(a, a)=1$, which is called reflexive.
(3) $s(a, b) = s(b, a)$, which is called symmetric.
(4) If $s(a, b) = 1$ and $s(b, c) = 1$ are held, one has $s(a, c) = 1$, which is called transitive.

Please refer to Subsection I.A in the supplementary material for the proofs of these observations.

C. Dividing decision variables based on interval similarity

For a DI-MOP, a well-designed algorithm is generally expected to have good performance in convergence and diversity before the problem changes. For a solution to a DI-MOP, some of its components change over dynamic parameters, and the others remain unchanged. If we divide all the decision variables of a DI-MOP into two groups, and optimize decision variables in each group individually, the efficiency of solving the optimization problem will be improved to some degree. In [58], a differential grouping strategy is employed to discover the underlying interaction structure of the decision variables. However, the strategy cannot be directly utilized to detect the interaction between a decision variable and a parameter in a DI-MOP, especially when the parameter is an interval, due to its different characteristics with a real value. For example, $Y = [1, 2]$ is an interval, $Y - Y = [-1, 1]$, not $[0, 0]$. Inspired by the study in [58], we propose a variant of the above method based on the interval similarity in this study, with the purpose of efficiently tackling a DI-MOP.

Definition 4: separable function. A function, $f(\mathbf{x}, \mathbf{c})$, is partially additively separable, if it has the following form:

$$
f(\mathbf{x}, \mathbf{c}) = \sum_{j=1}^{J} f_j(\mathbf{x}^j, \mathbf{c}^j)
$$

(5)

where $f_j(\mathbf{x}^j, \mathbf{c}^j)$ is a function associated with $\mathbf{x}^j$ and $\mathbf{c}^j$, $\mathbf{x}^1, \cdots, \mathbf{x}^J$ and $\mathbf{c}^1, \cdots, \mathbf{c}^J$ are mutually exclusive decision vectors and parameter vectors.

Theorem 1 For an additively separable interval function,

$f(\mathbf{x}, \mathbf{c}(t))$, its difference, denoted as $\Delta f(\mathbf{x}, \mathbf{c})$, will be as follows when x_k has a disturbance of δ.

$$\Delta f(\mathbf{x}, \mathbf{c}) = [\underline{\Delta f(\mathbf{x}, \mathbf{c})}, \overline{\Delta f(\mathbf{x}, \mathbf{c})}], \text{where}$$
$$\underline{\Delta f(\mathbf{x}, \mathbf{c})} = \underline{f(\cdots, x_k + \delta, \cdots, \mathbf{c})} - \underline{f(\cdots, x_k, \cdots, \mathbf{c})},$$
$$\overline{\Delta f(\mathbf{x}, \mathbf{c})} = \overline{f(\cdots, x_k + \delta, \cdots, \mathbf{c})} - \overline{f(\cdots, x_k, \cdots, \mathbf{c})}. \tag{6}$$

$\forall a, [\underline{c_p}, \overline{c_p}] \neq [\underline{c'_p}, \overline{c'_p}], \delta \in R, \delta \neq 0$, if the interval similiarity

$$s(\Delta f(\mathbf{x}, \mathbf{c})\big|_{x_k=a, c_p=[\underline{c_p}, \overline{c_p}]}, \Delta f(\mathbf{x}, \mathbf{c})\big|_{x_k=a, c_p=[\underline{c'_p}, \overline{c'_p}]}) < 1$$

is held, x_k and $c_p(c_p \in \mathbf{c})$ are said inseparable.

Please refer to Subsection I.B in the supplementary material for the detailed proof.

The grouping theory in [23] is a particular case as the parameter, $\mathbf{c}$, is a real vector in the above theory.

In [23] and [58], only one point is utilized to group the decision variables. However, the interval is characterized by two points, the left and the right endpoints. For interval optimization problems, if only one point is employed, information provided by the other endpoint will be lost, which results in inaccuracy. Taking a function, $f(x, c(t)) = c(t)x$, as an example, assume that $c(t) = [1, 1+t]$ and $\delta = 0.5$. On this circumstance, $\Delta_1 = \Delta f(x, c(t))\big|_{x=3, c_p=[1,2]} = [0.5, 1]$, $\Delta_2 = \Delta f(x, c(t))\big|_{x=3, c_p=[1,3]} = [0.5, 1.5]$. If only the left endpoints are utilized to measure the difference between Δ_1 and Δ_2, we will conclude that there is no difference according to the results of [23] and [58]. In fact, their right endpoints have a large difference. Nonetheless, based on the method proposed in this paper, their interval similarity is $s(\Delta_1, \Delta_2) = 0.5$, suggesting that Δ_1 and Δ_2 are clearly different. As a consequence, the method proposed in this paper can easily distinguish Δ_1 and Δ_2, and is more suitable for handling interval optimization problems.

In this study, due to uncertainties and noises generally existing in the objectives, we regard x_k and c_p inseparable if

$$s(\Delta f(\mathbf{x}, \mathbf{c})\big|_{x_k=a, c_p=[\underline{c_p}, \overline{c_p}]}, \Delta f(\mathbf{x}, \mathbf{c})\big|_{x_k=a, c_p=[\underline{c'_p}, \overline{c'_p}]}) < \theta_1$$

, where θ_1 is a threshold set in advance in the range of (0,1).

According to the above theorem, we propose the following approach of dividing all the decision variables in (1). For x_k and $c_p(t)$, we first calculate $\Delta f(\mathbf{x}, \mathbf{c}(t))\big|_{x_k=a, c_p(t)=[\underline{c_p}(t), \overline{c_p}(t)]}$ and $\Delta f(\mathbf{x}, \mathbf{c}(t))\big|_{x_k=a, c_p(t)=[\underline{c'_p}(t), \overline{c'_p}(t)]}$ according to (6). Following that, we obtain the similarity between the two intervals by (4), and if it is smaller than θ_1, x_k is inseparable with $c_p(t)$. On this circumstance, we delete x_k from the set of decision variables, and put it into the first group. The above process is repeated until all the interval parameters are detected. In this way, we can obtain decision variables in the first group. θ_1 is set according to noises and uncertainties existing in a problem, and the larger the noises or uncertainties, the smaller the value of θ_1 should be set. For a problem, θ_1 is generally set to 1. In fact, assigning its appropriate value is difficult. Algorithm 2 provides the pseudo code of the proposed method of dividing all the decision variables.

Algorithm 2: Dividing decision variables

Input: the time scale, t; upper and lower bounds of decision variables, *upb* and *lowb*; the number of objectives, decision variables and interval parameters, m, n and P, respectively;

Output: the groups of decision variables, $g1$ and $g2$;

1   $g1 \leftarrow \emptyset$ % $g1$ saves decision variables inseparable with $\mathbf{c}(t)$;
2   $g2 \leftarrow \emptyset$ % $g2$ saves decision variables separable with $\mathbf{c}(t)$;
3   **for** $i = 1$ *to* m **do**
4   **for** $p=1$ *to* P **do**
5   **for** $k =1$ *to* n **do**
6         $x(1:n) \leftarrow lowb(1:n)$;
7         $y(1:n) \leftarrow x(1:n)$;
8   % x and y mean the values of decision vector, $\mathbf{x}$;
9         $y(k) \leftarrow 0.5 \cdot (lowb(k) + upb(k))$;
10        $\mathbf{c}(1:P) \leftarrow \mathbf{c}(t)(1:P)$;
11        $\Delta1 \leftarrow f_i(x, \mathbf{c}) - f_i(y, \mathbf{c})$;
12        $\mathbf{c'}(1:P) \leftarrow \mathbf{c}(1:P)$;
13        $\mathbf{c'}(p) \leftarrow 0.5 \cdot \mathbf{c}(p)$;
14        $\Delta2 \leftarrow f_i(x, \mathbf{c'}) - f_i(y, \mathbf{c'})$;
15   Calculate the similarity between $\Delta1$ and $\Delta2$, s_{pk};
16   **if** $s_{pk} < \theta_1$ **then**
17          $g1 \leftarrow g1 \cup \{k\}$;
18   **end**
19   **end**
20   **end**
21   **end**
22   $g2 \leftarrow \{1, 2, \cdots, n\} \backslash g1$;
23   **return** $g1, g2$

Using the proposed strategy, all the decision variables can be divided into the following two groups: one is $\mathbf{x}^1 = (x_1^1, x_2^1, \cdots, x_r^1)$, which is inseparable with interval parameters, and the other is $\mathbf{x}^2 = (x_1^2, x_2^2, \cdots, x_{n-r}^2)$, which is separable with interval parameters.

D. Detecting problem change based on interval similarity

For a DI-MOP, if traditional approaches [25], [59], [60] are employed to detect the problem change, a predefined value associated with an objective interval, such as the left endpoint, the right endpoint, and the midpoint, is selected. However, the predefined value may not be changed as the problem varies. Let us consider the following three functions: $f_1(x, t) = [1, 1 + t]x$, $f_2(x, t) = [-t, 1]x$, $f_3(x, t) = [-t, t]x$, $(t > 1)$. When the parameter, t, varies, the left endpoint of $f_1(x, t)$, the right endpoint of $f_2(x, t)$, and the midpoint of $f_3(x, t)$ remain unchanged for the same value of x. Under this circumstance, traditional approaches have a difficulty in tackling a DI-MOP.

For a solution to a DI-MOP, at least one of its objective intervals will generally vary when the optimization problem changes, and the larger the change intensity of the optimization problem, the smaller the similarity between objective intervals of the previous and current optimization problems. In this way, we can detect whether the optimization problem change or not. To fulfill the task, we first form a detection population at the end of each generation, which is composed of a number of individuals chosen from the current optimal solutions. Then, we calculate the similarity between objective intervals of each individual in the detection population based on the previous and current optimization problems, and obtain the average interval similarity of these individuals. Finally, we compare the average interval similarity with a threshold, $\theta_2(0 < \theta_2 < 1)$, set in advance. If it is smaller than θ_2, the

---

Algorithm 3: Environment detection

Input: the archive, A; the number of objectives, m;
Output: the results of environment detection, *flag*;

1   Choose u individuals from the current optimal solutions to form the detection population, $det\_pop = \{p_1, p_2, \cdots, p_u\}$;
2   $i \leftarrow 1$;
3   **while** $i <= m$ **do**
4   **for** $j = 1$ *to* u **do**
5          $a_{ij}(t) = f_i(p_j, \mathbf{c}_i(t))$, $a_{ij}(t+1) = f_i(p_j, \mathbf{c}_i(t+1))$;
6   % Calculate the objective intervals based on the previous and the current optimization problems;
7   Calculate the similarity between $a_{ij}(t)$ and $a_{ij}(t+1)$, s_{ij};
8   **end**
9      $\bar{s}_i \leftarrow \frac{1}{u} \sum\limits_{j=1}^{u} s_{ij}$;
10   **if** $\bar{s}_i < \theta_2$ **then**
11        $flag \leftarrow$ true;
12   Break;
13   **else**
14        $flag \leftarrow$ false;
15        $i \leftarrow i + 1$;
16   **end**
17   **end**
18   **return** *flag*;

TABLE I
COMPARISON RESULTS WHEN DIFFERENT STRATEGIES ARE EMPLOYED TO DETECT THE ENVIRONMENTAL CHANGE

Function	Function value when $t = 1$	Function value when $t = 1.5$	Difference obtained by the traditional method	Similarity obtained by the proposed method
$f_1(2, t)$	[2, 4]	[2, 5]	0	0.6667
$f_2(2, t)$	[2.2, 4]	[2.3, 5]	0.1	0.6296
$f_3(2, t)$	4	5	1	0.8
$f_3(5, t)$	10	12.5	2.5	0.8

optimization problem is detected to have changed. θ_2 is set by a decision-maker according to his/her preference. The more sensitive the decision-maker to the environmental change, the larger the value of θ_2 should be set. On this circumstance, the decision-maker will frequently change his/her plan once the environment changes, and has a high requirement to the algorithm so as to rapidly seeking optimal solutions. The pseudo code of the proposed method of detecting the problem change is supplied in Algorithm 3.

To compare the traditional and the proposed detection strategies, three functions, $f_1(x, t) = (1, 1 + t)x$, $f_2(x, t) = (1 + 0.1t, 1 + t)x$, and $f_3(x, t) = (1 + t)x$, are taken into account. Among them, the first two are intervals, and the third one is a real value. Assume that only the left endpoints are employed in the traditional detection strategy. Table I lists the comparison results when the two strategies are employed to detect the environmental change. From this table, we can see:

(1) for interval functions, $f_1(x, t)$ and $f_2(x, t)$, rows 2 and 3 demonstrate that the traditional method is low-efficient, or even useless to detect the environmental change, due to its focus on only one point and neglect of the other one. However, the proposed method detects the environment change on the interval similarity, and hence successfully detect the changes with high-efficiency.

(2) for real function, $f_3(x, t)$, both methods have a capability in detecting the environmental change. However, the proposed method is less impacted by the decision variable, suggesting its robustness. Therefore, the proposed method is also suitable for real functions.

E. Responding to problem change

Once a change of the optimization problem is detected, appropriate strategies should be employed to respond to the change. Multi-population co-evolutionary optimization can efficiently search for the feasible decision space in multiple directions and interact with each other, which makes it suitable for addressing DMOPs. Compared with traditional single population evolutionary optimization, the multi-population counterpart generally has good performance in convergence and diversity. If all the populations respond to the change in the same way, the characteristics of different populations will be ignored, which deteriorates the performance of multi-population co-evolutionary optimization. A well-designed responding strategy is generally expected to have good performance in convergence and diversity. On the one hand, abandoning historical optimal solutions and randomly initializing a population is beneficial to the population diversity but may be time-consuming for an algorithm to converge. On the other hand, utilizing all the historical optimal solutions is helpful to converge, but may mislead the evolutionary direction when the change is severe. Furthermore, it generally results in the loss of the population diversity and falls into local optima. Therefore, seeking appropriate strategies for responding to the change is of considerable necessity.

To fulfill this task, we present a strategy for responding to changes of the optimization problem. In the proposed strategy, we initialize sub-populations corresponding to different groups using different methods. For decision variables in groups $\mathbf{x}^1$ and $\mathbf{x}^2$, they are optimized by two populations P^1 and P^2, respectively. For P^1, it is composed of two parts: one contains individuals which are randomly initialized in the feasible decision space to promote the diversity of P^1, and the other contains those which are initialized by a prediction strategy based on the change intensity to rapidly tracking the optimal solutions to the changed optimization problem. In the following, we will detail the prediction strategy.

1) Predicting the change direction of optimal solutions : To fulfill this task, we re-evaluate solutions in $A(t)$, the archive of complete solutions at the time scale t, and store non-dominated solutions in $A_0(t + 1)$. Then, the change direction can be obtained as follows.

$$D(t+1) = \frac{C_{A_0(t+1)} - C_{A(t)}}{\left\| C_{A_0(t+1)} - C_{A(t)} \right\|_2} \quad (7)$$

where $C_{A_0(t+1)}$ and $C_{A(t)}$ are the centroids of $A_0(t+1)$ and $A(t)$ in the decision space, respectively.

2) Estimating the step size of the change of optimal solutions: We define the change intensity of the optimization problem as follows.

$$\overline{ds}(t+1) = \frac{1}{m |A_0(t+1)|} \sum_{i=1}^{m} \sum_{j=1}^{|A_0(t+1)|} (1 - s_{ij}) \quad (8)$$

where $|A_0(t+1)|$ is the size of $A_0(t+1)$, s_{ij} refers to the similarity between the i-th objective of the j-th individual for the previous and current optimization problems, $\overline{ds}(t+1)$ is the average dissimilarity of each objective of all the individuals. $\overline{ds}(t+1)$ reflects the change intensity of the optimization problem, and a larger value of $\overline{ds}(t+1)$ indicates a more violent change.

Using historical information of optimal solutions, the step size of the change of optimal solutions is estimated as follows.

$$S(t+1) = \frac{\overline{ds}(t+1)}{\overline{ds}(t)} \left\| C_{A(t)} - C_{A(t-1)} \right\|_2 \qquad (9)$$

where $\frac{\overline{ds}(t+1)}{\overline{ds}(t)}$ means the ratio of the change intensity from time scale t to t+1, and a larger value of $\frac{\overline{ds}(t+1)}{\overline{ds}(t)}$ suggests that the change intensity at the time scale t+1 is stronger than that at time t. As a result, the step size of optimal solutions, $S(t+1)$, will be larger.

3) Generating new individuals: For any $p(t) \in P^1(t)$, its location for the changed optimization problem is predicted as follows.

$$p(t+1) = p(t) + rS(t+1)D(t+1) + \varepsilon(t) \qquad (10)$$

where r refers to a random value obeying the uniform distribution in [0,1]. $\varepsilon(t) \sim N\left(0, \sigma^2(t)I\right)$ is a Gaussian noise which is added to increase the probability of the re-initialized population to cover the PS of the changed optimization problem, I is an identity matrix, and $\sigma(t)$ is the standard deviation of the Gaussian distribution with its expression being $\sigma(t) = \frac{S(t+1)}{2\sqrt{n}}$.

P^1 is heavily affected by interval parameters when the optimization problem changes. According to this characteristic, the proposed response method utilizes the interval similarity of individuals fitness before and after the change to generate the initial solutions. Due to taking full advantage of information associated with the change, the generated initial solutions can well reflect the change trend of optimal solutions, leading to a rapid convergence.

With respect to P^2, interval parameters have no influences on $\mathbf{x}^2$. As a result, a half of individuals in P^2 are randomly initialized, and the rest are initialized with the Gaussian mutation, shown in (11), when the optimization problem changes.

$$p(t+1) = p(t) + \varepsilon(t) \qquad (11)$$

F. Constructing a complete solution

When evaluating an individual of the current population, forming a complete solution by selecting a representative individual from the other population together with the current individual is of necessity. To fulfill this task, the method in [23] is adopted to construct and evaluate a complete solution. For more details, please refer to [23].

G. Complexity of the proposed algorithm

In each generation, the main difference between the proposed CC-IP-MOEA-IS, which is achieved by combining the proposed cooperative co-evolutionary optimization and the response strategies based on the interval similarity with IP-MOEA, and IP-MOEA 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 two sub-populations with their size of $N/2$ to cooperative co-evolve when tackling an

optimization problem with $m(\geq 2)$ objectives, n decision variables, and p interval parameters. In additional, we assume that there are k non-separable variables for each interval parameter, and let n_co be the number of representatives from the other sub-populations during the evolution, and N be the sizes of the archive. The computational complexity associated with each of the above strategies is given as follows:
(1) grouping the decision variables is $O(mp(n+k))$ [58],
(2) detecting the environmental change is $O(mN)$ in the worst case,
(3) responding to the change is $O(mN)$,
(4) cooperatively co-evolving all the sub-populations is $O(m(n\_co)^2 N)$, and
(5) forming the archive is $O((2N)^m)$ [24], [61].

The number of representatives, n_co, is generally a small constant and irrelevant to the size of a sub-population. Therefore, the overall complexity of the proposed algorithm is $O((2N)^m)$, which is the same as that of the state-of-the-art IP-MOEA [61]. From this viewpoint, CC-IP-MOEA-IS is computationally efficient. However, if n_co is set to a large value, and relevant to the size of a sub-population, CC-IP-MOEA-IS will have a high computational complexity when $m = 2$.

IV. EXPERIMENTS

To evaluate the proposed algorithm, we conduct the following three groups of experiments. The first group investigates the influences of different response strategies by comparing the response strategy proposed in Section III. E with strategies A and B proposed in [25]. For the second group, its aim is to demonstrate the influences of cooperative co-evolution on an optimization algorithm. To fulfill this task, we compare IP-MOEA [24] with and without this paradigm when solving benchmark optimization instances. With the last group, we attempt to conduct a comprehensive comparison between the proposed algorithm and other five state-of-the-art ones. The implementation environment is provided as follows: Intel(R) Xeon(R) E5-2660 V3 CPU, 2.60GHz, 48GB RAM, Windows 10, MATLAB R2012a.

A. Benchmark optimization problems

To evaluate the proposed algorithm, eight new optimization problems, $ZDT3_{DI}$, $FDA1_{DI}$, $FDA2_{DI}$, $FDA4_{DI}$, $FDA5_{DI}$, and $DSW1_{DI}$-$DSW3_{DI}$, are constructed by modifying previous deterministic counterparts, ZDT3, FDA1, FDA2, FDA4, FDA5 [62] , and DSW1-DSW3 [63]. These problems can be scaled to any number of decision variables, and have concave, disconnected, scalable, and changeable Pareto fronts/sets. The first decision variable of each deterministic optimization problem is multiplied by an interval, $c_1 = [0.9, 1]$, and the rest are multiplied by intervals associated with the time scale, $c_i(t) = [0.45 \left| sin(0.5i\pi t) \right|, 0.5 + 0.45 \left| sin(0.5i\pi t) \right|]$ or $c_i(t) = [0.45 \left| sin(0.5\pi t) \right|, 0.5 + 0.45 \left| sin(0.5\pi t) \right|]$. In this way, the deterministic optimization problems can be converted into their interval counterparts. $c_i(t)$ changes over time, and different decision variables have different interval coefficients, which poses a difficulty to an algorithm when tackling these

optimization problems. Therefore, these optimization problems are qualified to test the capacity of an algorithm in tracking changing optimal solutions.

Please refer to Section II in the supplementary material for the detailed descriptions of them.

B. Compared algorithms and parameter settings

1) Compared algorithms and strategies: Since each sub-population in the proposed algorithm evolves using operators in IP-MOEA, it is necessary to take IP-MOEA as a compared algorithm. To investigate the efficiency of the response strategy proposed in this study, we take two state-of-the-art strategies, A and B, proposed by Deb et al. [25], into consideration for a comparative study. The three corresponding dynamic evolutionary optimization algorithms, CC-IP-MOEA-A, and CC-IP-MOEA-B, are achieved by combining the proposed cooperative co-evolutionary optimization, and the response strategies, A and B, respectively, with IP-MOEA. In addition, the other two algorithms, D-IP-MOEA-A and D-IP-MOEA-B, are obtained by incorporating the response strategies, A and B, respectively, with IP-MOEA. Finally, these two algorithms are compared with CC-IP-MOEA-A and CC-IP-MOEA-B to evaluate the performance of the proposed cooperative co-evolutionary strategy.

2) Parameter settings: In the experiments, the simulated binary crossover (SBX) and polynomial mutation are utilized with their distribution indexes of 20 to generate new offspring during the evolution. The crossover and mutation probabilities are 0.9 and $1/n$, respectively, where n is the number of decision variables. For each benchmark optimization problem, $t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$, where n_t, τ_t and τ are the change severity and frequency, as well as the maximal number of iterations, respectively. For problems ZDT3$_{DI}$, FDA1$_{DI}$, FDA2$_{DI}$, FDA4$_{DI}$, and FDA5$_{DI}$, $n_t = 10, \tau_t = 50, \tau = 2500$, indicating that there are fifty changes in total for each optimization problem. Each function is evaluated 10,000 times after the optimization problem changes. For problems DSW1$_{DI}$-DSW3$_{DI}$, $n_t = 2, \tau_t = 100, \tau = 2000$, i.e. twenty changes are tackled for each algorithm. Each function is evaluated 20,000 times after the change occurs.

At each generation, cooperative co-evolutionary algorithms, i.e., CC-IP-MOEA-A, CC-IP-MOEA-B, and CC-IP-MOEA-IS, have more evaluations than IP-MOEA. Therefore, the population size is set to 50 for these cooperative co-evolutionary algorithms and 200 for IP-MOEA, D-IP-MOEA-A, and D-IP-MOEA-B, to balance their total budget in evaluation. The archive size is set to 100, and the cooperative individual rate is set to 2. Besides, the values of θ_1 and θ_2 are set according to the tolerability of a decision-maker to the problem change and the robustness of obtained optimal solutions. In this study, they are both set to 0.9. For problems with their decision variables being inseparable with interval parameters, such as ZDT3$_{DI}$, FDA1$_{DI}$, and FDA2$_{DI}$, the decision variables are equally divided into two groups.

We run each algorithm 30 times for each optimization problem independently, and calculate the mean and standard deviation of the two performance indicators which will be formulated with (12) and (13). Besides, Mann-Whitney U test is adopted to show the difference of different algorithms in terms of each performance metric at the significant level of 5%.

C. Performance indicators

When evaluating a solution set obtained by an algorithm in convergence, diversity, and uncertainty, Limbourg and Aponte et al. [24] introduced such indicators as hyper-volume (*H*) and imprecision (*I*). To adapt the *H* and *I* indicators to DI-MOPs, the average value of each indicator in a period of time scales is calculated.

Definition 5: Average hyper-volume (*AH*). The value of the *AH* metric assists in assessing the tracking ability of a Pareto optimal set obtained by an algorithm before the optimization problem changes, with its expression as follows.

$$AH = [\underline{AH}, \ \overline{AH}] = \frac{1}{|T|} \sum_{t \in T} H(X^*(t)) \qquad (12)$$

where T is a set of time scales with its cardinality being $|T|$, $X^*(t)$ means the obtained PS(*t*) at the time scale, *t*. The larger the value of AH of the final Pareto front is, the closer the final front is to the true one, and the better the distribution of solutions along the front is. In the experiments, the reference point is set to $(1.2 f_1^{\max}, 1.2 f_2^{\max})$ for DSW3$_{DI}$ and $(5, 5)$ for the rest problems, where $f_1^{\max}$ and $f_2^{\max}$ are the maximal objectives of f_1 and f_2, respectively.

Definition 6: Average imprecision (*AI*). For $X^*(t)$, its imprecision is calculated as follows.

$$I(X^*(t)) = \sum_{x \in X^*(t)} \sum_{i=1}^{m} (\overline{f_i}(x, c_i(t)) - \underline{f_i}(x, c_i(t))).$$

The average imprecision of $X^*(t)$ is then calculated as follows.

$$AI = \frac{1}{|T|} \sum_{t \in T} I(X^*(t)). \qquad (13)$$

AI reflects the uncertainty of a Pareto optimal set in the objective space, and the smaller the value of *AI* of a Pareto optimal set, the more exact the true PF(*t*) corresponding to the Pareto optimal set.

D. Experimental results and discussion

In this subsection, the proposed algorithm, CC-IP-MOEA-IS, is compared with five algorithms, IP-MOEA, D-IP-MOEA-A, D-IP-MOEA-B, CC-IP-MOEA-A, and CC-IP-MOEA-B. Figs. 1 and 2 depict the boxplots of the *H* and *I* indicators obtained by different algorithms when tackling the eight DI-MOPs. Tables II and III list the experimental results, where data are the average/standard deviation of *AH*, *AI* or time consumption, the boldface ones are the best among these algorithms, and those labeled by '*' indicate that results obtained by the proposed algorithm are significantly different from those obtained by a comparative one.
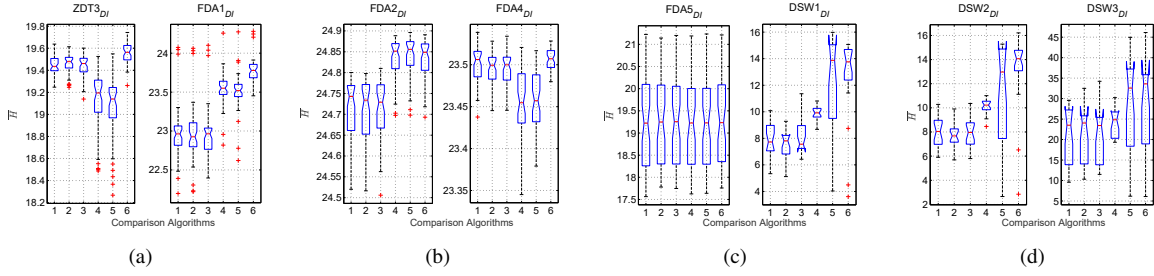
Fig. 1. The boxplot of $\overline{H}$ over 30 times of six algorithms on eight benchmark optimization problems. (1: IP-MOEA, 2: D-IP-MOEA-A, 3: D-IP-MOEA-B, 4: CC-IP-MOEA-A, 5: CC-IP-MOEA-B, 6: CC-IP-MOEA-IS.)
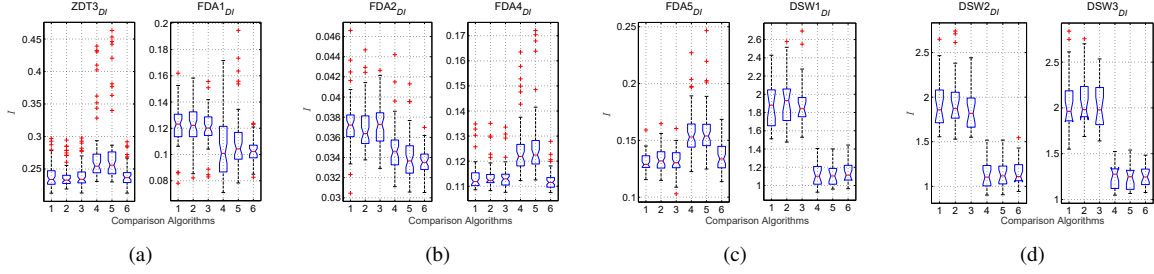

Fig. 2. The boxplot of I over 30 times of six algorithms on eight benchmark optimization problems. (1: IP-MOEA, 2: D-IP-MOEA-A, 3: D-IP-MOEA-B, 4: CC-IP-MOEA-A, 5: CC-IP-MOEA-B, 6: CC-IP-MOEA-IS.)

1) The performance of the proposed response strategy:
We first compare the two algorithms, D-IP-MOEA-A and D-IP-MOEA-B, on the eight benchmark problems to show the performance of incorporating response strategies A and B with IP-MOEA. Fig. 1 demonstrates that D-IP-MOEA-A and D-IP-MOEA-B are not better than IP-MOEA in terms of $\overline{H}$. Especially, the $\overline{H}$ values of the former are slightly smaller than that of the latter on $FDA4_{DI}$, $DSW1_{DI}$, and $DSW2_{DI}$, indicating that there is no significant improvement in convergence and diversity after incorporating response strategies A and B into IP-MOEA. In addition, there is no significant difference in terms of *AI* among the three algorithms, suggesting that response strategies A and B have slightly influence on the performance of an improved algorithm. We therefore conclude the unsuitability of response strategies A and B for DI-MOPs.

Then, we compare the algorithms, CC-IP-MOEA-A and CC-IP-MOEA-B with CC-IP-MOEA-IS, which have the same cooperative co-evolution paradigm and different response strategies, A, B, and interval similarity-based, to demonstrate the performance of the three response strategies. For $ZDT3_{DI}$, $FDA1_{DI}$, and $FDA4_{DI}$, CC-IP-MOEA-IS is significantly superior to CC-IP-MOEA-A and CC-IP-MOEA-B in terms of $\overline{H}$. When tackling problems $DSW1_{DI}$ and $DSW2_{DI}$, CC-IP-MOEA-IS achieves larger values of $\overline{H}$ than CC-IP-MOEA-A. Although CC-IP-MOEA-IS and CC-IP-MOEA-B have no significant difference in terms of the medium of $\overline{H}$, CC-IP-MOEA-IS has the $\overline{H}$ values with a smaller fluctuation than CC-IP-MOEA-B, which highlights its strong robustness. Moreover, on $DSW3_{DI}$, there is no significant difference in terms of $\overline{H}$ among CC-IP-MOEA-IS, CC-IP-MOEA-A, and CC-IP-MOEA-B. To sum up, the proposed response strategy achieves a larger $\overline{H}$ value and a smaller fluctuation on the other problems except $FDA2_{DI}$. The $\overline{AH}$ value in Table II also confirms the above observation, suggesting that the proposed

response strategy has a better capability to be combined with the cooperative co-evolution paradigm to promote the performance in convergence and diversity of an algorithm.

Additionally, CC-IP-MOEA-IS has not only smaller *I* values and stronger robustness than CC-IP-MOEA-A and CC-IP-MOEA-B in terms of *I* on $ZDT3_{DI}$, $FDA1_{DI}$, $FDA2_{DI}$, $FDA4_{DI}$, and $FDA5_{DI}$, but also its imprecise is as small as CC-IP-MOEA-A and CC-IP-MOEA-B on $DSW1_{DI}$-$DSW3_{DI}$.

From the above experimental results and analysis, we can conclude that the proposed response strategy has effectively improved in convergence and diversity of an algorithm. In addition, it significantly reduces the imprecise of the obtained optimal solution set. Hence, the proposed response strategy is more suitable for DI-MOPs.

2) The performance of the cooperative co-evolutionary paradigm: We compare the following pairs of algorithms, D-IP-MOEA-A and CC-IP-MOEA-A, D-IP-MOEA-B and CC-IP-MOEA-B, with each pair having the same response strategy, but different evolutionary paradigm. Fig. 1 demonstrates that, CC-IP-MOEA-A and CC-IP-MOEA-B are worse than their counterparts in terms of $\overline{H}$ on $ZDT3_{DI}$ and $FDA4_{DI}$. In addition, there is a slight difference among them on $FDA5_{DI}$. However, CC-IP-MOEA-A and CC-IP-MOEA-B have significantly larger $\overline{H}$ values than their counterparts on the rest five test instances, suggesting that the cooperative co-evolutionary paradigm can achieve good performance in convergence and diversity on most test cases.

We have the following observations from Fig. 2: CC-IP-MOEA-A and CC-IP-MOEA-B have larger *I* values than their counterparts on $ZDT3_{DI}$, $FDA4_{DI}$, and $FDA5_{DI}$, but their *I* values are generally smaller than their counterparts on the rest five problems, indicating that the cooperative co-evolutionary paradigm can also improve the imprecise of an algorithm.
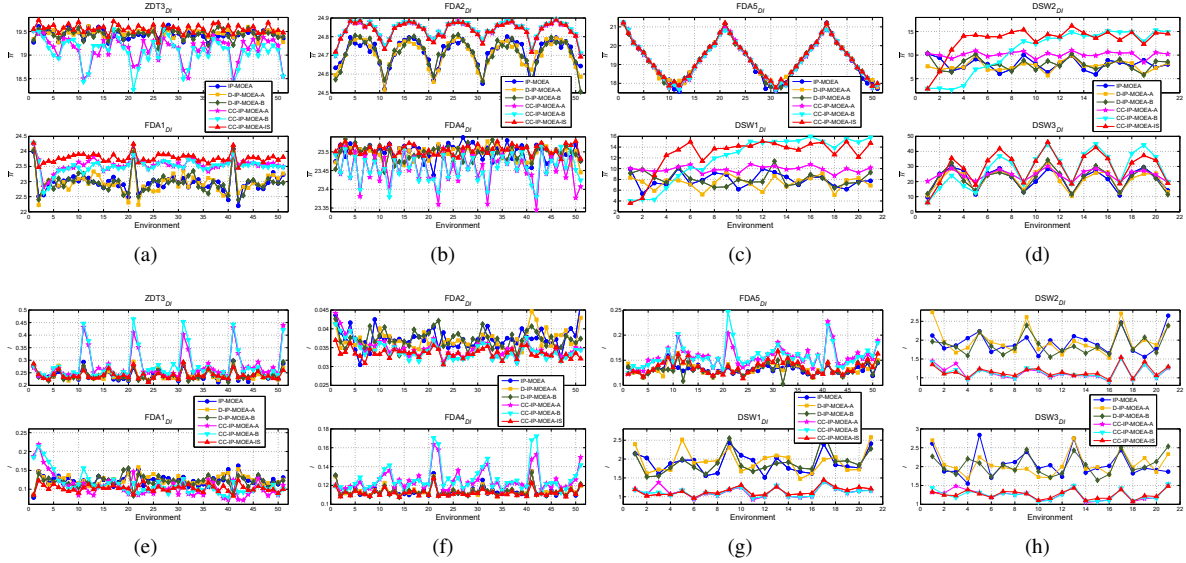
Fig. 3. The $\overline{H}$ and I values over 30 runs versus the time instances on eight benchmark optimization problems

E. Capability in tracking the time-variant Pareto fronts

The curves of $\overline{H}$ (i.e. the upper endpoint of H) and I over 30 runs versus the time instances on the eight benchmark problems are depicted in Fig. 3. From this figure, we can obtain:

(1) For ZDT3$_{DI}$, FDA4$_{DI}$, and FDA5$_{DI}$, two CCEAs, i.e. CC-IP-MOEA-A and CC-IP-MOEA-B, show no better performance in both convergence and diversity than the three non-CCEAs. Nevertheless, the proposed algorithm, CC-IP-MOEA-IS, obtains the competitive $\overline{H}$ and I values along with the change of an optimization problem, indicating its excellence in tracking time-dependent Pareto fronts. Furthermore, the proposed algorithm has the $\overline{H}$ and I values with a slighter fluctuation than the others, which highlights its strong robustness.

(2) For FDA1$_{DI}$ and FDA2$_{DI}$, three CCEAs are superior to the others no matter how an optimization problem changes, with good performance in robustness. In addition, CC-IP-MOEA-IS, which incorporates with the proposed response strategy, achieves the best performance in terms of $\overline{H}$ and I all the time. These results reveal that CC-IP-MOEA-IS is more suitable for handling dynamic interval problems.

(3) For DSW1$_{DI}$, DSW2$_{DI}$ and DSW3$_{DI}$, which have large feasible regions, CC-IP-MOEA-IS achieves the competitive $\overline{H}$ values, and is significant better than the non-CCEAs, except the first two time instances. Moreover, there is no significant difference in terms of the I values among the three CCEAs.

From the above results, we can conclude that the proposed algorithm is capable of rapidly tracking time-variant Pareto fronts as well as achieving a Pareto optimal set with good performance in convergence, diversity, and imprecision.

F. Comparison analysis of algorithms

Tables II and III list the averages and standard deviations of $\overline{AH}$ and AI obtained by different algorithms on the above

eight test instances. We have the following observations in terms of $\overline{AH}$ from Table II.

TABLE II
PERFORMANCE COMPARISONS OF DIFFERENT ALGORITHMS IN TERMS OF $\overline{AH}$

Problem	IP-MOEA	D-IP-MOEA-A	D-IP-MOEA-B	CC-IP-MOEA-A	CC-IP-MOEA-B	CC-IP-MOEA-IS
ZDT3$_{DI}$	19.4401*	19.4608*	19.4406*	19.1255*	19.0730*	**19.5584**
	0.0968	0.0871	0.1022	0.2706	0.3014	**0.0935**
FDA1$_{DI}$	22.9704*	22.9702*	22.9629*	23.5519*	23.5085*	**23.7838**
	0.3438	0.3686	0.347	0.2096	0.2366	**0.1568**
FDA2$_{DI}$	24.7172*	24.7095*	24.7144*	24.8358	**24.8406**	24.8329
	0.0727	0.0726	0.0753	0.0461	**0.0458**	0.0482
FDA4$_{DI}$	23.4994	23.4956*	23.4961*	23.4545*	23.4582*	**23.5064**
	0.0212	0.019	0.0194	0.042	0.0365	**0.013**
FDA5$_{DI}$	19.2303	19.2834	19.2472	19.2267	19.2302	**19.2845**
	1.0593	0.9789	1.029	1.0087	1.0053	**0.999**
DSW1$_{DI}$	7.9119*	7.4199*	8.0830*	9.9035*	11.9776	**12.6802**
	1.3337	1.1801	1.304	0.5761	4.1513	**3.2384**
DSW2$_{DI}$	8.0367*	7.7305*	8.0613*	10.1284*	10.9399	**13.1849**
	1.3064	1.0618	1.2896	0.633	4.627	**3.0862**
DSW3$_{DI}$	21.6077*	21.3705*	22.0210*	24.3956	28.6406	**29.6948**
	6.883	6.5181	7.042	3.3777	11.6022	**10.3569**

TABLE III
PERFORMANCE COMPARISONS OF DIFFERENT ALGORITHMS IN TERMS OF AI

Problem	IP-MOEA	D-IP-MOEA-A	D-IP-MOEA-B	CC-IP-MOEA-A	CC-IP-MOEA-B	CC-IP-MOEA-IS
ZDT3$_{DI}$	0.2397	**0.2397**	0.2415	.2779*	.2812*	0.2412
	-0.0202	**-0.0192**	-0.0206	-0.0563	-0.0642	-0.0171
FDA1$_{DI}$	0.1224*	0.1228*	0.1209*	0.1061	0.1107	**0.1025**
	0.0156	0.0154	0.014	0.0239	0.0232	**0.0089**
FDA2$_{DI}$	0.0375*	0.0370*	0.0373*	0.0349*	0.0341	**0.0336**
	0.0027	0.0024	0.0023	0.0024	0.0023	**0.0013**
FDA4$_{DI}$	0.1139	0.1140*	0.1139	0.1254*	0.1272*	**0.1123**
	0.0058	0.0053	0.0052	0.0116	0.0144	**0.0039**
FDA5$_{DI}$	**0.1308**	0.1334	0.1313	0.1568*	0.1581*	0.1359
	0.0083	0.0113	0.011	0.0208	0.0227	0.0131
DSW1$_{DI}$	1.9086*	1.9448*	1.9085*	1.1283	**1.1184**	1.1461
	0.2777	0.3031	0.2973	0.1345	**0.1169**	0.115
DSW2$_{DI}$	1.9313*	1.9834*	1.8851*	1.1439	**1.1392**	1.1602
	0.2813	0.3501	0.273	0.1734	**0.1604**	0.1506
DSW3$_{DI}$	2.0628*	2.0640*	2.0393*	1.2564	**1.2458**	1.2587
	0.3462	0.3029	0.2723	0.1458	**0.1419**	0.1132

(1) With the help of the cooperative co-evolutionary paradigm and the response strategy, CC-IP-MOEA-IS significantly outperforms the other five in terms of $\overline{AH}$ on ZDT3$_{DI}$ and FDA1$_{DI}$. Taking ZDT3$_{DI}$ as an example, it is Type I, and its true PF(t) is discontinuous. CC-IP-MOEA-IS has the $\overline{AH}$ value of 19.5584, which is better than

IP-MOEA (19.4401), D-IP-MOEA-A(19.4408), D-IP-MOEA-B(19.4406), CC-IP-MOEA-A(19.1255), and CC-IP-MOEA-B (19.0730), indicating that CC-IP-MOEA-IS has better performance in convergence and distribution.

(2) For FDA2$_{DI}$, it is Type II, and its PS(t) and PF(t) vary as the optimization problem changes. As a result, it is difficult to be tracked when the optimization problem changes. The proposed algorithm, CC-IP-MOEA-IS, is clearly superior to IP-MOEA, D-IP-MOEA-A, and D-IP-MOEA-B. Although the $\overline{AH}$ value of CC-IP-MOEA-IS (24.8329) is slightly smaller than those of CC-IP-MOEA-A (24.8358) and CC-IP-MOEA-B (24.8406), there is no significant difference between CC-IP-MOEA-IS and each of CC-IP-MOEA-A and CC-IP-MOEA-B.

(3) For FDA4$_{DI}$, CC-IP-MOEA-IS performs the best in terms of $\overline{AH}$, and is significantly better than the other four except IP-MOEA. For FDA5$_{DI}$, its PS(t) and PF(t) change over time, and its convexity varies, suggesting that it is difficult for an algorithm to rapidly tracking the changing PF(t). On this test case, although there is no significant difference in terms of $\overline{AH}$ between the proposed algorithm and the rest, CC-IP-MOEA-IS obtains the best $\overline{AH}$ value.

(4) For DSW1$_{DI}$-DSW3$_{DI}$, there is no significant difference between CC-IP-MOEA-IS and CC-IP-MOEA-B in terms of $\overline{AH}$. However, CC-IP-MOEA-IS is clearly superior to the other four and achieves the largest value of $\overline{AH}$ among all the comparative algorithms. Therefore, the proposed algorithm has a strong capacity in tracking the optimal solutions to DSW1$_{DI}$-DSW3$_{DI}$.

Furthermore, we have the following observations in terms of *AI* from Table III.

(1) Although CC-IP-MOEA-IS does not achieve the minimal *AI* value on ZDT3$_{DI}$ and FDA5$_{DI}$, there is no significant difference between CC-IP-MOEA-IS and each of IP-MOEA, D-IP-MOEA-A, and D-IP-MOEA-B. In addition, CC-IP-MOEA-IS outperforms CC-IP-MOEA-A and CC-IP-MOEA-B in terms of the *AI* indicator. Taking FDA$_{DI}$5 as an example, although the *AI* value of the proposed algorithm, 0.1359, is slightly bigger than those of IP-MOEA (0.1308), D-IP-MOEA-A (0.1334), and D-IP-MOEA-B (0.1313), there is no significant difference. Furthermore, it is clearly smaller than those of. CC-IP-MOEA-A (0.1568) and CC-IP-MOEA-B (0.1581).

(2) For FDA2$_{DI}$, CC-IP-MOEA-IS is clearly better than the others but CC-IP-MOEA-B. Moreover, CC-IP-MOEA-IS has also achieved the smallest *AI* value in the six compared algorithm on FDA4$_{DI}$.

(3) CC-IP-MOEA-A, CC-IP-MOEA-B, and CC-IP-MOEA-IS have no significant difference in terms of *AI* on FDA1$_{DI}$, DSW1$_{DI}$-DSW3$_{DI}$. They have smaller *AI* values than the other three algorithms, suggesting that the algorithms incorporating with the cooperative co-evolutionary paradigm gain the smallest imprecision.

Based on the above experimental results and analyses, we have the following conclusions..

(1) Using an appropriate response strategy is beneficial to improving the performance of EAs for DI-MOPs. For example, D-IP-MOEA-A and D-IP-MOEA-B are not significantly better than IP-MOEA in terms of $\overline{AH}$ and *AI* on most test cases.

However, CC-IP-MOEA-IS is superior to CC-IP-MOEA-A and CC-IP-MOEA-B, indicating that the proposed response strategy is more suitable for DI-MOPs than response strategies A and B. The main reason is that the proposed response strategy can rapidly respond to the change of an optimization problem by accurately predicting the new location of the evolutionary population.

(2) The cooperative co-evolutionary paradigm is not always effective. Although CC-IP-MOEA-A and CC-IP-MOEA-B are good at addressing FDA1$_{DI}$, FDA2$_{DI}$, DSW1$_{DI}$-DSW3$_{DI}$, they do not work on ZDT3$_{DI}$ and FDA5$_{DI}$.

(3) Appropriately combining the strategies proposed in this paper can improve the performance of an EA. CC-IP-MOEA-IS, which is generated by incorporating the proposed response strategy and CC into IP-MOEA, has the best performance among all the six algorithms. The reason is that the proposed response strategy has the capability in rapidly responding to the change of an optimization, and cooperative co-evolutionary paradigm is good at speeding up convergence. Therefore, the algorithm proposed in this paper is competitive when solving DI-MOPs.

In addition, experiments about time consumption for each comparative algorithm and the effectiveness of the archive set are conducted in Section III of Supplementary Material. From the experimental results, we can conclude that each IP-MOEA with cooperative co-evolution has significantly longer time consumption than the others. Additionally, the proposed algorithm obtains the largest $\overline{H}$ value when the archive size is 100 and 150. For more details, please refer to Section III of Supplementary Material.

V. APPLICATION IN A MULTI-PERIOD PORTFOLIO SELECTION PROBLEM

In this section, we investigate a multi-period portfolio selection problem in emerging markets [64], [65]. To provide investors with more choices, we formulate the problem with uncertainties as a bi-objective optimization model with interval coefficients in the objectives. In the formulated model, the expected return rate and risk loss rate at the tth period are represented as follows:

$$R(x, r(t)) = \sum_{i=1}^{n} [\underline{r_{t,i}}, \overline{r_{t,i}}] x_{t,i} - \sum_{i=1}^{n} a_{t,i} |x_{t,i} - x_{t-1,i}| + r_{t,0} x_{t,0},$$

$$Q(x, r(t), q(t)) = \sum_{i=1}^{n} \left([\underline{q_{t,i}}, \overline{q_{t,i}}] + \frac{1}{2(\overline{r_{t,i}} - \underline{r_{t,i}})} (r_{t,i} - \underline{r_{t,i}})^2 \right) x_{t,i}.$$

Therefore, the interval bi-objective optimization model can be formulated as

$$\min \ (-R(x, r(t)), Q(x, r(t), q(t)))$$
$$\text{s.t.} \ \sum_{i=1}^{n} x_{i,t} = 1, \ x_{i,t} \geq 0, \ t = 1, 2, ..., T. \tag{14}$$

where $r(t), q(t), a_{t,i}$ are parameters, with their meaning and setting being found in [64], [65]. In addition, $r(t), q(t)$ are intervals.

According to Theorem 1, $x_{t,0}$ is one separable variable and the others are inseparable with interval parameters, $r(t)$ and/or $q(t)$, in model (14). The proposed algorithm, CC-IP-MOEA-IS, and the comparative ones are employed to tackle the optimization problem.

Each algorithm runs 20 times independently. We save these results and calculate their means. The reference point is set to (0, 0.2) when computing hyper-volume. The values of the best hyper-volume and time consumption are listed in Table IV. In addition, we depict the best result obtained by CC-IP-MOEA-IS, IP-MOEA, and CC-IP-MOEA-B, with $t=2$ and $t=3$ in Fig. 4, respectively, to intuitively demonstrate the advantages of the proposed algorithm.

TABLE IV
PERFORMANCE OF DIFFERENT ALGORITHMS ON THE PORTFOLIO
SELECTION PROBLEM

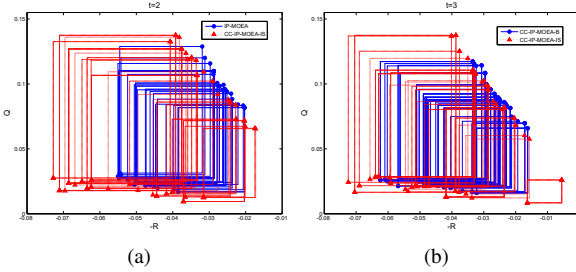| Algorithm	$\overline{H}$			Time consumption(s)		
	$t=1$	$t=2$	$t=3$	$t=1$	$t=2$	$t=3$
IP-MOEA	0.0037	0.0039*	0.0039*	**0.3198**	**0.3205**	**0.3204**
D-IP-MOEA-A	0.0037	0.0038*	0.0037*	0.3784	0.3737	0.3728
D-IP-MOEA-B	0.0037	0.0038*	0.0035*	0.4405	0.439	0.439
CC-IP-MOEA-A	**0.0039**	0.0040*	0.0040*	0.6419	0.6365	0.6377
CC-IP-MOEA-B	**0.0039**	0.0042*	0.0041*	0.6405	0.639	0.639
CC-IP-MOEA-IS	0.0038	**0.0044**	**0.0045**	0.6746	0.6723	0.668

Fig. 4. PFs obtained by different algorithms on the portfolio selection problem

Table IV reports that although the proposed algorithm, CC-IP-MOEA-IS, does not achieve the maximal $\overline{H}$ value at $t=1$, there is no significant difference between CC-IP-MOEA-IS and each comparative algorithm. Furthermore, CC-IP-MOEA-IS reaches the best $\overline{H}$ values among all the algorithms at $t=2$ and $t=3$, despite its time consumption is the longest, indicating that CC-IP-MOEA-IS has the best performance in convergence and distribution. From Fig. 4, we can conclude that the proposed algorithm can archive more and better solutions than IP-MOEA and CC-IP-MOEA-B. Taking Fig. 4a as an example, on the one hand, CC-IP-MOEA-IS offers investors a larger return rate with the same risk loss rate than IP-MOEA. On the other hand, CC-IP-MOEA-IS provides investors more choices with different preferences.

VI. CONCLUSIONS

Focusing on DI-MOPs with time-varying interval parameters, we have proposed a cooperative co-evolutionary algorithm, termed CC-IP-MOEA-IS, by incorporating the interval similarity-based grouping strategy and response strategy into IP-MOEA. In CC-IP-MOEA-IS, all the decision variables are first divided into two groups, interrelated with/without interval parameters, according to the interval similarity-based grouping strategy. Then, two sub-populations are utilized to evolve those groups, with the search space of each sub-population being shrunk, and the capability in tracking the optimal solutions being prompted. At the end of each generation, the interval similarity of objectives is adopted to detect whether the optimization problem changes or not. Once a change is detected,

the two groups are re-initialized according to the proposed response strategy. To evaluate CC-IP-MOEA-IS, we have employed it to address eight benchmark optimization cases provided in Table II in comparison with five algorithms and two response strategies. The experimental results demonstrate that CC-IP-MOEA-IS, the interval similarity-based grouping and the proposed response strategies are very competitive among the comparative algorithms and strategies on most optimization instances whose decision variables are separable with interval parameters. In addition, the proposed algorithm and strategies have better performance on optimization instances whose decision variables are inseparable with interval parameters, such as $ZDT3_{DI}$, $FDA1_{DI}$, and $FDA2_{DI}$.

It is worth noting that we have evaluated the proposed algorithm only on a few benchmark optimization instances, and investigated its scalability by applying it to a practical optimization problem. In addition, the two sub-populations utilized to optimize the two groups have the same size, which consumes many computing resources in searching for optimal solutions of a group which is weakly impacted by the changing interval parameters. For these problems, new efficient methods are required to explore, which will be the focus of our future work.

REFERENCES

[1] J. Lin, M. Liu, J. Hao, and S. Jiang, "A multi-objective optimization approach for integrated production planning under interval uncertainties in the steel industry," *Computers and Operations Research*, vol. 72, pp. 189–203, 2016.

[2] Y. Zhang, D. Gong, and J. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, no. 2, pp. 172–185, 2013.

[3] E. Zhang and Q. Chen, "Multi-objective reliability redundancy allocation in an interval environment using particle swarm optimization," *Reliability Engineering and System Safety*, vol. 145, pp. 83–92, 2016.

[4] H. T. Nguyen, L. B. Le, and Z. Wang, "A bidding strategy for virtual power plants with intraday demand response exchange market using stochastic programming," *IEEE Transactions on Industry Applications*, vol. 54, no. 4, pp. 3044–3055, 2018.

[5] A. Bozorgi-Amiri, M. Jabalameli, and S. Al-E-Hashem, "A multi-objective robust stochastic programming model for disaster relief logistics under uncertainty," *OR Spectrum*, vol. 35, no. 4, pp. 905–933, 2013.

[6] Z. Sazvar, S. Al-E-Hashem, A. Baboli, and M. Jokar, "A bi-objective stochastic programming model for a centralized green supply chain with deteriorating products," *International Journal of Production Economics*, vol. 150, no. 6, pp. 140–154, 2014.

[7] M. S. Pishvaee and J. Razmi, "Environmental supply chain network design using multi-objective fuzzy mathematical programming," *Applied Mathematical Modelling*, vol. 36, no. 8, pp. 3433–3446, 2012.

[8] M. S. Moreno and A. M. Blanco, "A fuzzy programming approach for the multi-objective patient appointment scheduling problem under uncertainty in a large hospital," *Computers and Industrial Engineering*, vol. 123, pp. 33–41, 2018.

[9] D. Kumar, Z. Rahman, and F. Chan, "A fuzzy AHP and fuzzy multi-objective linear programming model for order allocation in a sustainable supply chain: A case study," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 6, pp. 535–551, 2017.

[10] Y. Hong, J. Lin, C. Wu, and C. Chuang, "Multi-objective air-conditioning control considering fuzzy parameters using immune clonal selection programming," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1603–1610, 2012.

[11] D. Gong, J. Sun, and Z. Miao, "A set-based genetic algorithm for interval many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 47–60, 2018.

[12] S. Nojavan and K. Zare, "Interval optimization based performance of photovoltaic/wind/fc/electrolyzer/electric vehicles in energy price determination for customers by electricity retailer," *Solar Energy*, vol. 171, pp. 580–592, 2018.

[13] L. Wang and C. Xiong, "Sequential multidisciplinary design optimization and reliability analysis under interval uncertainty," *Aerospace Science and Technology*, vol. 80, pp. 508–519, 2018.

[14] X. Sun, D. Gong, Y. Jin, and S. Chen, "A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 685–698, 2013.

[15] D. Gong, N. Qin, and X. Sun, "Evolutionary algorithms for optimization problems with uncertainties and hybrid indices," *Information Sciences*, vol. 181, no. 19, pp. 4124–4138, 2011.

[16] A. K. Bhunia and S. S. Samanta, "A study of interval metric and its application in multi-objective optimization with interval objectives," *Computers and Industrial Engineering*, vol. 74, no. 1, pp. 169–178, 2014.

[17] L. Sahoo, A. K. Bhunia, and P. K. Kapur, "Genetic algorithm based multi-objective reliability optimization in interval environment," *Computers and Industrial Engineering*, vol. 62, no. 1, pp. 152–160, 2012.

[18] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *IEEE Congress on Evolutionary Computation*. IEEE, 2001, pp. 1101–1108.

[19] M. Potter and K. Jong, "A cooperative coevolutionary approach to function optimization," *Lecture Notes in Computer Science*, vol. 866, pp. 249–257, 1994.

[20] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[21] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2018.

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

[23] B. Xu, Y. Zhang, D. Gong, Y. Guo, and M. Rong, "Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1877–1890, 2018.

[24] P. Limbourg and D. E. S. Aponte, "An optimization algorithm for imprecise multi-objective problem functions," in *The IEEE Congress onEvolutionary Computation*, vol. 1, 2005, pp. 459–466.

[25] K. Deb, N. Rao, and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2007, pp. 803–817.

[26] Y. Liu, M. Li, H. Lian, X. Tang, C. Liu, and C. Jiang, "Optimal dispatch of virtual power plant using interval and deterministic combined optimization," *International Journal of Electrical Power and Energy Systems*, vol. 102, pp. 235–244, 2018.

[27] L. Majumder and S. S. Rao, "Interval-based optimization of aircraft wings under landing loads," *Computers and Structures*, vol. 87, no. 3, pp. 225–235, 2009.

[28] Z. Zhao, X. Han, C. Jiang, and X. Zhou, "A nonlinear interval-based optimization method with local-densifying approximation technique," *Structural and Multidisciplinary Optimization*, vol. 42, no. 4, pp. 559–573, 2010.

[29] L. Wang and C. Xiong, "Interval number optimization for household load scheduling with uncertainty," *Energy and Buildings*, vol. 130, pp. 613–624, 2016.

[30] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, 2009.

[31] Z. Q. Cheng, "Feasibility analysis for optimization of uncertain systems with interval parameters," *Acta Automatica Sinica*, vol. 30, no. 3, pp. 455–459, 2004.

[32] C. Jiang, X. Han, F. Guan, and Y. Li, "An uncertain structural optimization method based on nonlinear interval number programming and interval analysis method," *Engineering Structures*, vol. 29, no. 11, pp. 3168–3177, 2007.

[33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[34] D. Gong, N. Qin, and X. Sun, "Evolutionary algorithms for multi-objective optimization problems with interval parameters," in *IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications*, 2010, pp. 411–420.

[35] X. Sun, D. Gong, and W. Zhang, "Interactive genetic algorithms with large population and semi-supervised learning," *Applied Soft Computing Journal*, vol. 12, no. 9, pp. 3004–3013, 2012.

[36] J. Sun and D. Gong, "Solving interval multi-objective optimization problems using evolutionary algorithms with lower limit of possibility degree," *Chinese Journal of Electronics*, vol. 22, no. 2, pp. 269–272, 2013.

[37] Y. Zhang, D. Gong, G. Hao, and Y. Jiang, "Particle swarm optimization for multi-objective systems with interval parameters," *Acta Automatica Sinica*, vol. 34, no. 8, pp. 921–928, 2008.

[38] C. K. Goh and K. C. Tan, "Handling noise in evolutionary multi-objective optimization," *Evolutionary Multi-objective Optimization in Uncertain Environments: Issues and Algorithms, Springer Berlin Heidelberg*, pp. 55–99, 2009.

[39] H. Karshenas, C. Bielza, and P. Larranaga, "Interval-based ranking in noisy evolutionary multi-objective optimization," *Computational Optimization and Applications*, vol. 61, no. 2, pp. 517–555, 2015.

[40] R. Dou, C. Zong, and M. Li, "An interactive genetic algorithm with the interval arithmetic based on hesitation and its application to achieve customer collaborative product configuration design," *Applied Soft Computing*, vol. 38, no. C, pp. 384–394, 2016.

[41] J. Sun, D. Gong, X. Zeng, and N. Geng, "An ensemble framework for assessing solutions of interval programming problems," *Information Sciences*, vol. 436-437, pp. 146–161, 2018.

[42] Y. Chen, X. Y. Sun, D. W. Gong, Y. Zhang, J. Choi, and S. Klasky, "Personalized search inspired fast interactive estimation of distribution algorithm and its application," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 588–600, 2017.

[43] R. Shang, L. Jiao, M. Gong, and W. Ma, "An immune clonal algorithm for dynamic multi-objective optimization," *Journal of Software*, vol. 18, no. 11, pp. 2700–2711, 2007.

[44] R. Shang, J. Wang, L. Jiao, and Y. Wang, "An improved decomposition-based memetic algorithm for multi-objective capacitated arc routing problem," *Applied Soft Computing Journal*, vol. 19, no. 1, pp. 343–361, 2014.

[45] R. Azzouz, S. Bechikh, and L. B. Said, "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy," *Soft Computing*, vol. 21, no. 4, pp. 885–906, 2017.

[46] W. T. Koo, K. G. Chi, and K. C. Tan, "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment," *Memetic Computing*, vol. 2, no. 2, pp. 87–110, 2010.

[47] R. Chen, K. Li, and X. Yao, "Dynamic multi-objectives optimization with a changing number of objectives," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, 2018.

[48] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2007, pp. 832–846.

[49] R. Gan, G. Yu, J. Zheng, J. Zou, and S. Yang, "The effect of diversity maintenance on prediction in dynamic multi-objective optimization," *Applied Soft Computing*, vol. 58, pp. 631–647, 2017.

[50] R. Liu, J. Fan, and L. Jiao, "Integration of improved predictive model and adaptive differential evolution based dynamic multi-objective evolutionary optimization algorithm," *Applied Intelligence*, vol. 43, no. 1, pp. 192–207, 2015.

[51] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via kalman filter prediction," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2016.

[52] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, "Multidirectional prediction approach for dynamic multiobjective optimization problems," *IEEE Transactions on Cybernetics*, in press, DOI:10.1109/TCYB.2018.2842158, 2018.

[53] R. Shang, L. Jiao, Y. Ren, J. Wang, and Y. Li, "Immune clonal coevolutionary algorithm for dynamic multiobjective optimization," *Natural Computing*, vol. 13, no. 3, pp. 421–445, 2014.

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

[55] C. Yue, B. Qu, and J. Liang, "A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 805–817, 2017.

[56] B. Qu, J. Liang, Y. S. Zhu, and P. N. Suganthan, "Solving dynamic economic emission dispatch problem considering wind power by multi-objective differential evolution with ensemble of selection method," *Natural Computing*, in press, DOI:10.1007/s11047-016-9598-6, 2017.

[57] R. Xu and Z. Xu, "On similarity degrees of interval numbers," *Mathematics in Practice and Theory (in Chinese)*, vol. 37, no. 24, pp. 1–8, 2007.

[58] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2014.

[59] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014.

[60] Y. Guo, Z. Yang, C. Wang, and D. Gong, "Cultural particle swarm optimization algorithms for uncertain multi-objective problems with interval parameters," *Natural Computing*, vol. 16, no. 4, pp. 1–22, 2016.

[61] D. Gong, J. Sun, and Z. Miao, "A set-based genetic algorithm for interval many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 47–60, 2018.

[62] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.

[63] J. Mehnen, T. Wagner, and G. Rudolph, "Evolutionary optimization of dynamic multi-objective test functions," *Sonderforschungsbereiche*, vol. 1, pp. 1–10, 2006.

[64] Y. J. Liu, W. G. Zhang, and J. B. Wang, "Multi-period cardinality constrained portfolio selection models with interval coefficients," *Annals of Operations Research*, vol. 244, no. 2, pp. 1–25, 2016.

[65] J. Sun, X. Yan, Z. Heng, and L. Zhiping, "Intervalmulti-objective programming methods for solving multi-period portfolio selection problems," *Control and Decision (in Chinese)*, in press, DOI:10.13195/j.kzyjc.2018.0574.
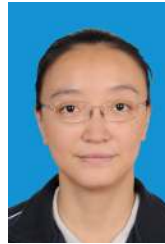
Yinan Guo received the PhD degree in control theory and control engineering from the China University of Mining and Technology, in 2003. She is a professor in the School of Information and Control Engineering, China University of Mining and Technology. Her research interests include intelligence optimization and control.

Shengxiang Yang (M'00–SM'14) received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China, in 1993, 1996, and 1999, respectively.

He is currently a Professor in Computational Intelligence and Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 270 publications. He has over 8100 Google Scholar citations with an H-index of 48. His current research interests include evolutionary computation, swarm intelligence, artificial neural networks, data mining and data stream mining, and relevant real-world applications. He serves as an Associate Editor/Editorial Board Member of a number of international journals, such as the IEEE Transactions on Evolutionary Computation, IEEE Transactions on Cybernetics, Information Sciences, Evolutionary Computation, and Soft Computing.

Dunwei Gong received the Ph.D. degree in control theory and control engineering from China University of Mining and Technology, Xuzhou, China, in 1999. He is a professor in the School of Information and Control Engineering, China University of Mining and Technology. His main research interests include intelligence optimization and control.

Biao Xu received the M.Sc. degree in Applied Mathematics from Guilin University of Electronic Technology, Guilin, China, in 2006. He is currently a Ph.D. candidate at the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China. His research interests include intelligence optimization and data mining.

Yong Zhang received the M.Sc. and Ph.D. degrees in control theory and control engineering from China University of Mining and Technology, Xuzhou, China, in 2006 and 2009, respectively. He is currently with the School of Information and Control Engineering, China University of Mining and Technology. His research interests include intelligence optimization and data mining.