High-Dimensional Robust Multi-Objective Optimization for Order Scheduling: A Decision Variable Classification Approach

Wei Du, *Member, IEEE*, Weimin Zhong, Yang Tang, *Member, IEEE*, Wenli Du, and Yaochu Jin, *Fellow, IEEE*

***Abstract*—This paper tackles the high-dimensional robust order scheduling problem. A multi-objective evolutionary algorithm called constrained nondominated sorting differential evolution based on decision variable classification is developed to search for robust order schedules. The decision variables are classified into highly and weakly robustness-related variables according to their contributions to the robustness of candidate solutions. The experimental results reveal that the performance of robust evolutionary optimization can be greatly improved via analyzing the properties of decision variables and then decomposing the high-dimensional robust optimization problem. It is also unveiled that the order scheduling is greatly affected by the uncertain daily production quantities. The robust order schedules are able to provide more information on earliness/tardiness of the orders, which enhances the flexibility of the production.**

***Index Terms*—Decision variable classification (DVC), evolutionary multi-objective optimization, high-dimensional optimization, robust evolutionary optimization, robust order scheduling.**

Manuscript received November 16, 2017; revised March 19, 2018 and April 27, 2018; accepted May 3, 2018. Date of publication May 15, 2018; date of current version January 3, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61703163, in part by Shanghai Sailing Program under Grant 17YF1427700, in part by the China Postdoctoral Science Foundation under Grant 2016M601525, in part by the Fundamental Research Funds for the Central Universities under Grant 222201714028, and in part by the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017. The work of Y. Jin was supported by EPSRC under Grant EP/M017869/1. Paper no. TII-17-2657. *(Corresponding authors: Weimin Zhong; Yang Tang.)*

W. Du, W. Zhong, Y. Tang, and W. Du are with the Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China (e-mail: duwei0203@gmail.com; wmzhong@ecust.edu.cn; tangtany@gmail.com; wldu@ecust.edu.cn).

Y. Jin is with the Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China, and also with the Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: yaochu.jin@surrey.ac.uk).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This consists of a PDF file containing additional information relevant to the main paper. This material is 0.292 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2836189

I. Introduction

In SUPPLY chain management, order scheduling is one of the vital decision-making problems when producing time-sensitive products. Take apparel industry as an example, manufacturers typically receive orders from retailers shortly before a selling season. The orders are subject to huge product varieties, which increases labor costs in the production process [1]. Therefore, order scheduling is of paramount importance to rational resource distribution, which benefits enterprises [2], [3].

In the recent two decades, order scheduling problems have been widely studied from different aspects. For example, the order scheduling problem was solved by considering order release, order sequencing, and group scheduling in a single-stage production system [4]. In [5], the schedules were obtained at the supply chain level with the consideration of the assignment of orders to/at each plant as well as the shipment schedule. In [6], the order scheduling problem was addressed under a complex manufacturing environment, which considers multiple processes, multiple departments, and multiple plants. For recent advances in order scheduling, the reader is referred to [7]–[10] and references therein.

Recently, as a powerful optimization tool [11]–[13], evolutionary algorithms (EAs) have been brought in to tackle order scheduling problems [6], [14], [15]. In these studies, researchers mainly made efforts in proposing new EAs. In addition, daily production quantities were assumed to be fixed on a specified production line. Nevertheless, in real production, there are multiple kinds of disruptions that often occur in the order scheduling [7], including tool failure, machine breakdown, and operator illness, among others. Hence, daily production quantities often vary during the production process. Under these circumstances, order schedules that are robust to the uncertain daily production quantities are preferred. The robust order scheduling problem belongs to robust optimization problems. In searching for a candidate for making robust order schedules, robust evolutionary optimization has been shown to be promising [16], [17]. Practitioners may control the desired level of robustness based on practical situations by setting different constraints in robust evolutionary optimization. As a result, the first motivation of this research comes from solving order scheduling problems by robust evolutionary optimization.

The task of order scheduling is to assign the orders received from retailers to proper production lines in terms of the delivery date of each order. An order scheduling problem belongs to a very complex combinatorial optimization problem, which has a massive solution space. For instance, for 20 orders and four production lines, the number of the candidate solutions is up to 4^{20}. When some real-world production factors (e.g., order split, learning effect, etc.) are considered, the encoding scheme of the order scheduling problem may become more complicated, and the scheduling becomes a high-dimensional problem with more than 100 decision variables [18]. In robust optimization, as the dimension of the decision space grows, the complexity of the problem's fitness landscape increases dramatically. Moreover, the robust region of the high-dimensional problem is much harder to determine when the number of the decision variables increases. This is known as the "curse of dimensionality" [19], which implies that the performance of robust optimization methods deteriorates as the dimensionality of the search space grows. It is worth mentioning that high-dimensional robust optimization, especially high-dimensional robust multi-objective optimization, has so far received little attention.

To solve high-dimensional single-objective optimization problems, cooperative coevolution (CC) is widely utilized [20]. The idea of CC is to decompose a high-dimensional optimization problem into a group of subproblems that can be separately optimized by conventional EAs. Two representative grouping mechanisms are random grouping [21] and differential grouping [22]. Recently, high-dimensional multi-objective optimization has attracted increasing attention. A novel grouping mechanism is proposed based on a decision variable analysis strategy, which investigates whether a decision variable contributes to convergence, diversity or both. Then, the decision variables are partitioned as convergence-related variables and diversity-related variables [23], or position variables, distance variables, and mixed variables [24]. Some promising experimental results are reported in [23] and [24]. Inspired by the variable property-based classification, high-dimensional robust multi-objective optimization problems can be handled by identifying whether a decision variable strongly influences the robustness of candidate solutions. Hence, the second motivation of this research aims to facilitate the solution of high-dimensional robust multi-objective optimization problems by categorizing the decision variables into different groups using a dedicated decision variable classification (DVC) based approach.

Based on the above discussion, this research addresses the high-dimensional robust order scheduling problem using a DVC-based approach. The order scheduling problem is formulated as a high-dimensional robust multi-objective optimization problem. A multi-objective evolutionary algorithm (MOEA) called constrained nondominated sorting differential evolution based on DVC (CNSDE/DVC) is developed to search for robust order schedules. In CNSDE/DVC, decision variables are classified into highly robustness-related variables and weakly robustness-related variables according to their contributions to the robustness of candidate solutions. Then, the two groups of the decision variables are optimized separately in CNSDE/DVC.

The contributions of this paper can be summarized as follows.

1) High-dimensional robust multi-objective optimization problems are addressed. To the best of our knowledge, existing work on robust optimization has been limited to low to medium-dimensional problems. For example, nondominated sorting genetic algorithm II (NSGA-II) was adopted for the robust engineering design [17], nondominated sorting composite differential evolution (NSCDE) was developed for robust multi-objective controllability of complex neuronal networks [25], and nondominated sorting adaptive differential evolution (NSJADE) was proposed for the robust multi-objective order scheduling in the discrete manufacturing industry [26]. The dimensions of the problems investigated in [17], [25], and [26] are 4, 10–100, and 80, respectively.
2) The decision variables are categorized into groups based on their influence on the robustness of the solutions, thereby enhancing the efficiency in solving high-dimensional problems.
3) The proposed algorithm is applied to solving high-dimensional robust order scheduling problems and compared with three state-of-the-art MOEAs for solving robust multi-objective optimization problems.

This paper is organized as follows. Section II formulates the robust order scheduling problem. Section III introduces the details of CNSDE/DVC including the DVC operation, as well as the CNSDE/DVC-based robust order scheduling. Section IV provides a set of experiments, as well as the experimental results. Finally, concluding remarks are given in Section V.

II. Problem Formulation

In this paper, we take the apparel industry as an example and discuss the robust order scheduling. The notations used in the problem are set based on Fast React [27], a business software specially developed for the apparel industry.

Nomenclature

Notations Related to Production Order

m	Total number of orders.
O_i	ith order ($1 \leq i \leq m$).
Q_i	Order size of O_i.
SD_i	Starting date of O_i in the schedule.
FD_i	Finishing date of O_i in the schedule.
DD_i	Due date of O_i.
EO_i	Efficiency of processing O_i.
PT_i	Processing time of O_i.
RT_i	Processing time that is required to achieve the next efficiency level for O_i.
CT_i	Time used on processing O_i on the current efficiency level.
S_i^{mins}	Standard time on producing each piece of O_i.
Q_{is}	Quantity of O_i that is completed on the sth day of producing O_i ($1 \leq s \leq FD_i - SD_i$).
$Q_{\text{sum,is}}$	Quantity of O_i finished from the first day to the sth day of producing O_i ($1 \leq s \leq FD_i - AD_i$).
r	Number of suborders divided for O_i.
β_i	Split percentage of O_i.

O_{iq} qth suborder of O_i $(1 \le q \le r)$.

Notations Related to Production Line

n Total number of production lines.

P_j ith production line $(1 \le j \le n)$.

C_j^{mins} Capacity each day of P_j.

EP_{jp} Efficiency of processing product of type T_p on P_j.

Other Notations

MD Day when the schedule is made.

PD Day when the production starts.

nt Number of product types.

T_p pth product type $(1 \le p \le nt)$.

U_p Consecutive working days of processing the product of T_p.

$f_p(\cdot)$ Function that denotes the learning effect of processing the order of T_p.

γ Uncertainty factor of daily production quantities.

The task of the order scheduling in the apparel industry is to assign m orders to n production lines appropriately. During the production, orders can be split for flexible production. In addition, the production lines belong to product-specific lines, which implies the production efficiency on a line can reach the highest only for certain type of product. Learning effect is also considered in the problem. The problem is described on the basis of [26]; because of the page limit, we provide the detailed problem description in Section S.I of the supplementary file.

An appropriate schedule implies that both earliness and tardiness of each order are discouraged [28], [29]. The reason is that the storage costs will increase (i.e., higher earliness penalty costs) when an order is completed before its due date, and the customer satisfaction will reduce (i.e., higher tardiness penalty costs) when an order is finished after its due date [28]. As a result, the two optimization objectives are set. First, minimizing the total earliness of all the orders, second, minimizing the total tardiness of all the orders.

In detail, the first objective is expressed as follows:

$$f_1 = \sum_{i=1}^{m} g_1(FD_i - DD_i) \tag{1}$$

where FD_i and DD_i are the finishing date and the due date of O_i in the schedule, respectively; and $g_1(\cdot)$ is

$$g_1(u) = \begin{cases} 0, & \text{if } u \ge 0 \\ -u, & \text{otherwise.} \end{cases} \tag{2}$$

The second objective is described as follows:

$$f_2 = \sum_{i=1}^{m} g_2(FD_i - DD_i) \tag{3}$$

where FD_i and DD_i are the finishing date and the due date of O_i in the schedule, respectively; and $g_2(\cdot)$ is

$$g_2(u) = \begin{cases} 0, & \text{if } u \le 0 \\ u, & \text{otherwise.} \end{cases} \tag{4}$$

These two objectives are conflicting, which implies a solution that results in a smaller f_1 (less total earliness) will lead to a larger f_2 (more total tardiness).

In this paper, the order scheduling considers uncertain daily production quantities, which affects FD_i of each order. Hence, the schedules obtained are robust to the variations of daily production quantities. Then, the formulation of robust multi-objective optimization will be introduced. Here, robust multi-objective optimization is formulated in terms of the second type of multi-objective robust solutions in [17], which controls the desired level of robustness according to practical situations via setting different constraints for the optimization problem. As a result, the robust multi-objective optimization problem can be formulated as

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\ \text{s.t.} \quad & \|\mathbf{f}^{\text{eff}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| \le \eta \\ & \mathbf{x} \in \Omega \end{aligned} \tag{5}$$

where $\mathbf{f}^{\text{eff}}(\mathbf{x}) = (f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), \ldots, f_M^{\text{eff}}(\mathbf{x}))$, and $f_i^{\text{eff}}(\mathbf{x})$ is defined as follows:

$$f_i^{\text{eff}}(\mathbf{x}) = \frac{1}{|\mathcal{B}_\delta(\mathbf{x})|} \int_{\mathbf{y} \in \mathcal{B}_\delta(\mathbf{x})} f_i(\mathbf{y}) d\mathbf{y} \tag{6}$$

where $\mathcal{B}_\delta(\mathbf{x})$ denotes a δ-neighborhood of $\mathbf{x}$, $|\mathcal{B}_\delta(\mathbf{x})|$ indicates the related hypervolume of the neighborhood; $f_i^{\text{eff}}(\mathbf{x})$ represents the ith $(1 \le i \le M)$ mean effective objective function. In (5), $\mathbf{x} = [x_1, x_2, \ldots, x_D]^T$ is a decision vector, the size of which is D; Ω denotes the feasible decision space; $\|\cdot\|$ can be any suitable norm; η is a constant that controls the desired level of robustness and the value is predefined by the practitioners.

Therefore, the order scheduling problem can be converted into a constrained bi-objective optimization problem

$$\begin{aligned} \text{minimize} \quad & f_1 = \sum_{i=1}^{m} g_1(FD_i - DD_i) \\ \text{minimize} \quad & f_2 = \sum_{i=1}^{m} g_2(FD_i - DD_i) \\ \text{s.t.} \quad & \|f_1^{\text{eff}} - f_1\|_1 + \|f_2^{\text{eff}} - f_2\|_1 \le \eta \end{aligned} \tag{7}$$

where L^1 norm is utilized in the constraint.

We hope that robust order schedules that balance the two objectives can be obtained based on the problem formulation. These schedules will provide early warnings of earliness or tardiness to the planners. Then, more warehouse spaces can be prepared as early as possible for the early orders, while more operators can be arranged to work extra time for the delayed orders. In the next section, we will give a detailed account of how to search for the robust order schedules.

Remark 1: It is worth mentioning that the apparel industry belongs to labor-intensive industries, which require a large amount of human labor to produce products. For order scheduling problems in the apparel industry, production uncertainties often arize due to operator absenteeism or operator illness. In addition, the production can be affected by operator's diverse efficiencies of producing different types of products (i.e., production lines are product-specific lines) and operators' increasing output efficiencies of continuously producing the same type of product (i.e., learning effect). Therefore, order scheduling

Algorithm 1: The Framework of CNSDE/DVC.

Begin
  /* *NP*: population size
  /* *POP*: current population
  /* *SN*: number of selected individuals for DVC
  /* *PN*: number of perturbations on each decision variable of selected individuals
  /* *DV1*: highly robustness-related variables
  /* *DV2*: weakly robustness-related variables
  /* c_1: number of cycle 1
  /* c_2: number of cycle 2
  POP = *Population_Initialization*(*NP*)
  [*DV1*, *DV2*] = *DVC_Operation*(*POP*, *SN*, *PN*)
  while the maximum evaluation number is not achieved **do**
    for $i_1 = 1 : c_1$ **do**
      POP = *DV1_Optimization*(*POP, DV1*)
    end for
    for $i_2 = 1 : c_2$ **do**
      POP = *DV2_Optimization*(*POP, DV2*)
    end for
  end while
end

Algorithm 2: *DVC_Operation*(*POP*, *SN*, *PN*).

1: **Begin**
2: /* *TN*: total number of the perturbation operation repeated on selected individuals
3: /* *D*: dimension size of decision variables
4: /* *POP*: current population
5: /* *SN*: number of selected individuals for DVC
6: /* *PN*: number of perturbations on each decision variable of selected individuals
7: /* *DV1*: highly robustness-related variables
8: /* *DV2*: weakly robustness-related variables
9: /* *VarCV*: size of $SN \times D$
10: /* *AvgVal*: size of $1 \times D$
11: /* *AllVal*: size of $TN \times D$
12: /* *TVal*: size of $1 \times D$
13: **for** $k = 1 : TN$ **do**
14: **for** $i = 1 : D$ **do**
15: Randomly select *SN* individuals from *POP*
16: **for** $j = 1 : SN$ **do**
17: Perturb *PN* times for the ith decision variable of the jth individual and record the variance of constraint violation values as $VarCV_{ji}$
18: **end for**
19: **end for**
20: Record the average value of *VarCV* for each decision variable in *AvgVal*
21: $AllVal(k,:)$=*AvgVal*
22: **end for**
23: $TVal = \text{sum}(AllVal(:,i) < \theta, 1)$
24: Find the decision variables that meet $TVal >$ mean $(TVal)$ and record as *DV2*
25: The rest decision variables are recorded as *DV1*
26: **end**

problems in the apparel industry can be modeled as complex robust optimization problems. For other industries, there are also multiple disruptions in the production such as machine breakdown or tool failure. Hence, the results obtained in this paper can also be extended to order scheduling problems in other industries.

III. Proposed Algorithm: CNSDE/DVC

In this section, the framework of CNSDE/DVC is first presented. Then, the details of the DVC operation and other main operations in CNSDE/DVC are elaborated. Finally, the CNSDE/DVC-based robust order scheduling is introduced.

A. Framework of CNSDE/DVC

The framework of CNSDE/DVC is listed in Algorithm 1. There are four components in CNSDE/DVC: *Population_Initialization*, *DVC_Operation*, *DV1_Optimization*, and *DV2_Optimization*. First, a population of *NP* individuals is initialized in a random way. Second, by means of the DVC operation, the decision variables are divided into two categories: highly robustness-related variables and weakly robustness-related variables. Then, the two categories of variables are repeatedly optimized for a certain number of cycles, respectively, until the maximum number of fitness evaluations is exhausted.

B. DVC Operation

As introduced in Section II, the robust optimization problem is converted into a constrained optimization problem. Thus, searching for robust solutions is equivalent to searching for feasible solutions. The DVC operation divides the decision variables into two groups: highly robustness-related variables and weakly robustness-related variables. A decision variable is called a highly/weakly robustness-related variable if the feasibility of the solution is highly/less sensitive to perturbations on the variable. In the following discussion, we use *DV1* and *DV2* to represent highly or weakly robustness-related variables, respectively.

The detailed procedure of the DVC operation is listed in Algorithm 2. For the DVC operation, the main idea lies in perturbing the decision variables, and then monitoring the changes to the constraint violation.

Lines 13–22 describe the repeated operation of perturbing each decision variable of a number of individuals. *SN* individuals are first randomly chosen from the population. Then, *PN* perturbations are carried out on each decision variable of the selected *SN* individuals, after which the variance values of the related constraint violation are recorded in *VarCV*. The size of *VarCV* is $SN \times D$, where D is the dimension size of decision variables. Finally, for each decision variable, the average value of *VarCV* for each decision variable is kept in *AvgVal*, the size of which is $1 \times D$. The above-mentioned operation is repeated for

TABLE I
EXAMPLE TO ILLUSTRATE HOW TO IDENTIFY *DV1* AND *DV2* AMONG x_1, x_2, x_3, AND x_4

Decision Variables	x_1	x_2	x_3	x_4
AvgVal value (the 1st time)	1.7500	1.0250	2.7250	0.2292
AvgVal value (the 2nd time)	0.5083	0.4917	0.1667	0.2292
AvgVal value (the 3rd time)	0.2500	0.2417	0.7917	0
AvgVal value (the 4th time)	1.2583	0.3083	3.5750	0.1250
AvgVal value (the 5th time)	1.6750	0.2083	0.0833	0.4792
TVal value	1	4	2	5

The *AvgVal* value that is less than the threshold θ is highlighted in grey background.

TN times, the purpose of which is to increase the accuracy of the classification since the characteristics of the decision variables can only be captured via *SN* randomly selected individuals. All the *TN* values of *AvgVal* are stored in *AllVal*, the size of which is $TN \times D$. The above-mentioned operation returns a group of average variance values of the constraint violation affected by the perturbation on each decision variable.

Lines 23–25 show the classification operation based on the results from the perturbation. First, a threshold θ is set, and we calculate for how many of the total *TN* times for each decision variable the average variance value of the constraint violation is smaller than θ. The values of the times are recorded in *TVal*, the size of which is $1 \times D$. Then, the decision variables that satisfy $TVal > \text{mean}(TVal)$ are classified as weakly robustness-related variables, i.e., *DV2*; the rest decision variables are classified as highly robustness-related variables, i.e., *DV1*.

In Table I, an example is given to show how to identify *DV1* and *DV2*. For an optimization problem, there are four decision variables x_1, x_2, x_3, and x_4. A number of *SN* individuals ($SN = 4$) are randomly picked out from the population for DVC. For each decision variable of these *SN* individuals, a number of *PN* perturbations ($PN = 8$) are conducted. Then, we calculate the average variance values of the constraint violation affected by the perturbation on x_1 to x_4. These operations are repeated for *TN* times ($TN = 5$). The related *AvgVal* values are recorded in the second to the sixth rows in Table I. The threshold θ is set as 0.5, and we count for how many of the total *TN* times for x_1 to x_4 (i.e., *TVal*) the average variance value of the constraint violation is smaller than θ. The *AvgVal* value that is less than θ is highlighted in grey background in Table I. The *TVal* value is recorded in the last row of Table I. Hence,

$$\text{mean}(TVal) = (1 + 4 + 2 + 5)/4 = 3. \tag{8}$$

Then, the decision variables that satisfy $TVal > \text{mean}(TVal)$ (i.e., x_2 and x_4) are classified as *DV2*; the rest decision variables (i.e., x_1 and x_3) are classified as *DV1*.

In a word, the DVC operation divides the decision variables into *DV1* and *DV2* via perturbing the decision variables, observing the changes to the constraint violation and comparing the changes with the predefined threshold.

C. Optimization of DV1 and DV2

After grouping the decision variables into two groups, *DV1* and *DV2*, CNSDE/DVC starts to optimize each category of decision variables separately. This research aims to propose a novel decomposition method for high-dimensional robust optimization problems, hence, we only use existing optimization strategies for *DV1* and *DV2*. Classical differential evolution (DE) DE/rand/1/bin is utilized to optimize *DV1* and *DV2*. Because of the page limit, the first two operations *Mutation* and *Crossover* are given in Section S.II of the supplementary file. While the *Selection* operation [13] is introduced below. It is worth pointing out that besides DE, other EAs like genetic algorithms or swarm-based intelligent algorithms like particle swarm optimization and ant colony optimization methods can also be used as the search engine in CNSDE/DVC.

Selection: *NP* offspring individuals are generated from *NP* parent individuals after mutation and crossover operations. Then, the offspring population is combined with the parent population, and a new population of *NP* individuals will be selected from the combined population.

The mutation and crossover operations are the same for the *DV1* and *DV2* optimization. While for the selection operation, the selection rules are different. *DV1* are decision variables that are highly related to robustness. It is desirable to obtain the solutions with high robustness by optimizing *DV1*. Therefore, robustness is used as the selection criterion for *DV1*, and individuals with higher robustness are preferred to enter the next generation. While for *DV2*, they are weakly related to robustness. We aim to enhance the convergence and diversity performance of the population by optimizing *DV2*. Hence, nondomination rank is utilized as the first selection criterion, and crowding distance is set as the second selection criterion. Individuals with a lower (better) nondomination rank and a larger crowding distance will be selected as parents of the next generation. The fast nondominated sorting is used to sort the population with a lower computational complexity compared to traditional methods [13].

In the optimization process, *DV1* and *DV2* are alternately optimized for c_1 and c_2 cycles, respectively, until the stopping criterion is reached.

Remark 2: It is worth mentioning that CNSDE/DVC is developed for the high-dimensional robust order scheduling problem, which is modeled as a high-dimensional robust multi-objective optimization problem. Unlike other existing MOEAs [17], [25], [26] that are used to solve robust multi-objective optimization problems, CNSDE/DVC makes it easier to solve the high-dimensional robust problem by classifying the decision variables into *DV1* and *DV2* and optimizing them separately. Detailed experimental results will be provided in Section IV.

D. CNSDE/DVC-Based Robust Order Scheduling

CNSDE/DVC is then used to optimize the robust order scheduling problem. Two important issues related to the problem need to be explained: *Encoding Scheme* and *Population Evaluation*.

1) Encoding Scheme: The task of the order scheduling is to assign m orders to n production lines properly. Potential order schedules should be encoded before the optimization. A potential solution should reflect how the orders are distributed on the production lines. Moreover, the en-

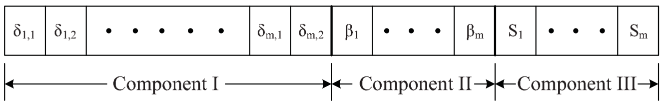


Fig. 1. Encoding scheme for the problem.

coding should also reveal how an order is divided and how the orders are assigned on a single production line. Therefore, a potential solution is composed of the following three components:

1) the order-line relationship;
2) the split details of each order; and
3) the arrangement details of the orders on a single line.

In real-world production, orders are not split frequently, so the maximum number of the suborders for each order is 2. As a result, the size of a potential solution is $D = 4m$, where m is the total number of the orders.

Fig. 1 illustrates the encoding scheme. In Component I, every two bits $\delta_{i,1}$ and $\delta_{i,2}$ $(1 \leq i \leq m)$ denote on which production line order O_i is assigned. The size of Component I is $2m$. In Component II, each single bit represents the split percentage per order. The size of Component II is m. In Component III, every single bit stores the label of each order, which implies how the orders are arranged on each production line. The size of Component III is m.

When initializing the population, each bit of Component I is initialized with a random real number that is uniformly selected from $(0, n]$, where n indicates the number of the production lines. Component II represents the split percentage when dividing each order. In this paper, the split percentage is chosen from the set $\{0.2, 0.4, 0.6, 0.8\}$. Each dimension of Component II is initialized with a random real number uniformly generated from $(0.1, 0.8]$. For the initialization of Component III, each bit is initialized with a random real number in $(0, m]$, where m is the total number of the orders.

It can be noticed that the encoding scheme is real-valued and continuous in the defined intervals. As stated in [30], with such an encoding scheme, classical real-coded EAs such as DE and PSO can be applied conveniently.

2) Population Evaluation: Before evaluating the population, we utilize the *ceil* operator to process the value of each bit in the encoding scheme. Specifically, the value of each bit in Component I is processed by $\lceil \delta_{i,1} \rceil$ and $\lceil \delta_{i,2} \rceil$; the value of each bit in Component II is processed by $\lceil \beta_i / 0.2 \rceil \cdot 0.2$; the value of each bit in Component III is processed by $\lceil S_i \rceil$. This operation acts as a bridge between the encoding scheme and the population evaluation. An illustration is provided in Section S.III of the supplementary file.

Then, according to the problem formulation in Section II, f_i and f_i^{eff} $(i = 1, 2)$ of each potential solution should be calculated. f_i^{eff} is the mean effective objective function. To compute $f_i^{\text{eff}}(\mathbf{x})$, H neighboring points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_H$ are generated around $\mathbf{x}$ based on Latin hypercube strategy. Then, the fitness values $f_i(\mathbf{x}_1), f_i(\mathbf{x}_2), \ldots, f_i(\mathbf{x}_H)$ can be calculated by (7). Finally, we average the values of $f_i(\mathbf{x}_1), f_i(\mathbf{x}_2), \ldots, f_i(\mathbf{x}_H)$, and $f_i^{\text{eff}}(\mathbf{x})$ is obtained.

After *Encoding Scheme* and *Population Evaluation* are settled, the robust order scheduling problem can be solved by CNSDE/DVC. In the following section, a set of numerical experiments are carried out to prove the effectiveness of CNSDE/DVC.

IV. Experimental Results and Analysis

A. Experimental Settings

To save space, we provide the experimental settings, which include the details of the experimental data in Section S.IV of the supplementary file.

We consider 40 orders in the experiments. Hence, for the problem, the dimension size is $D = 160$. The maximum number of function evaluations (MAX_FES) is set as $D \cdot 10\,000$. The population size is $NP = 100$. The scaling factor and the crossover probability of the DE algorithm are set as $F = 0.5$ and $CR = 0.9$, respectively. In the experiments, the uncertainty factor of daily production quantities is $\gamma = 0.3$. We set the number of the neighboring points for each potential solution as $H = 5$. The desired level of robustness for this problem is predefined as $\eta = 5$.

In the DVC operation, we need to first determine the value of *PN*. There are three components in the encoding scheme of the problem investigated in this paper. According to the value range of each component, *PN* is set as $PN = 6$ for Component I, $PN = 4$ for Component II, and $PN = 40$ for Component III. There are five parameters that should be discussed in CNSDE/DVC, and they are *SN*, *TN*, θ, c_1, and c_2. In the experiments, the settings are $SN = 4$, $TN = 15$, $\theta = 1$, $c_1 = 40$, and $c_2 = 8$. The parameter sensitivity study of these five parameters is listed in Section S.V of the supplementary file.

Each algorithm is run for 30 times. Two performance metrics inverted generational distance (IGD) and hypervolume (HV) are used to quantify all the experiments in comparison. To calculate IGD, a set of reference points need to be provided beforehand. In this paper, the nondominated solutions obtained from the combined solutions of all the algorithms under comparison are set as the reference points. To calculate HV, the maximum value of each objective over all the solutions multiplied by a constant 1.1 composes the reference point. We calculate the values of IGD and HV by using PlatEMO, which is a recently designed evolutionary multi-objective optimization platform [31]. Furthermore, in order to draw statistically sound conclusions, a Wilcoxon rank-sum test at a 0.05 significance level is conducted to evaluate the significance of the differences between the results obtained by two competing algorithms.

B. Comparison of CNSDE/DVC and CNSDE

One main contribution of this paper is to decompose decision variables according to their influence on the robustness of candidate solutions to more efficiently solve high-dimensional robust order scheduling problems. Therefore, we first examine the effectiveness of CNSDE/DVC by comparing it with CNSDE, in which the DVC operation is not considered.

TABLE II
PERFORMANCE COMPARISON BETWEEN CNSDE AND CNSDE/DVC

Algorithm	IGD values (mean±std)	HV values (mean±std)
CNSDE	71.88±6.77†	2.89E+05±2.99E+03†
CNSDE/DVC	**12.50±4.09**	**3.12E+05±3.22E+03**

The better results are highlighted. "†" indicates that the result of the peer algorithm is significantly different from that of CNSDE/DVC at a 0.05 level by the Wilcoxon rank-sum test.

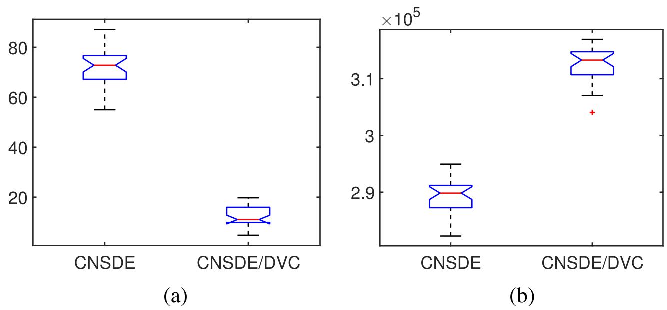


Fig. 2. Box plots for the IGD and HV values of CNSDE and CNSDE/DVC after 30 runs. (a) IGD. (b) HV.

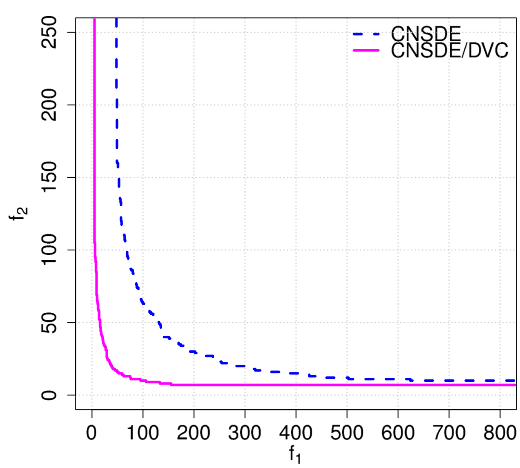


Fig. 3. Median attainment surfaces of 30 independent runs of CNSDE and CNSDE/DVC.

The IGD and HV values of CNSDE and CNSDE/DVC are listed in Table II. In addition, the box plots for the IGD and HV values of CNSDE and CNSDE/DVC after 30 runs are provided in Fig. 2. Moreover, the median attainment surfaces of 30 independent runs of CNSDE and CNSDE/DVC are also shown in Fig. 3 [32]. It can be observed that CNSDE/DVC greatly improves the performance of CNSDE when handling the high-dimensional robust order scheduling problem.

In CNSDE/DVC, the total 160 decision variables are divided into *DV1* (i.e., highly robustness-related variables) and *DV2* (i.e., weakly robustness-related variables). In Fig. 4, we provide the frequency of the decision variables, which are classified as *DV1* by CNSDE/DVC after 30 runs. It is observed from Fig. 4 that the decision variables from No. 81 to No. 120 are seldom identified as *DV1*. The reason can be inferred as follows: the 81st to the 120th decision variable indicates the split percentage of each order. Compared to other decision variables, varying these 40 decision variables will only affect the suborder size, and will not affect the order sequence on each single line. Keeping the arrangement of the orders unchanged in a schedule (the suborder size might be altered) indicates that the constraint violation keeps largely unchanged when perturbing each component of the decision variables from No. 81 to No. 120. Therefore, these 40 decision variables are most likely to be grouped into *DV2* instead of *DV1*.

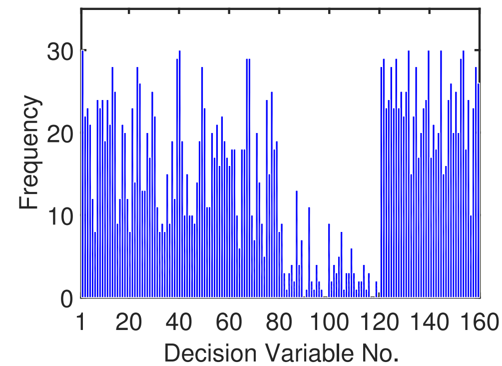


Fig. 4. Frequency of the decision variables, which are classified as *DV1* by CNSDE/DVC after 30 runs.

TABLE III
PERFORMANCE COMPARISON OF CNSDE/DVC, NSGA-II, NSCDE, AND NSJADE

Algorithm	IGD values (mean±std)	HV values (mean±std)
CNSDE/DVC	**11.74±4.19**	**7.72E+04±1.37E+03**
NSGA-II	135.92±20.31†	4.15E+04±4.43E+03†
NSCDE	17.72±5.94†	7.70E+04±1.15E+03
NSJADE	28.76±8.41†	7.39E+04±2.07E+03†

The best results are highlighted. "†" indicates that the result of the peer algorithm is significantly different from that of CNSDE/DVC at a 0.05 level by the Wilcoxon rank-sum test.

C. Comparison With Three MOEAs

Since high-dimensional robust multi-objective optimization problems have so far received little attention, no dedicated algorithms have been developed. Therefore, in the comparative studies, we select NSGA-II [17], NSCDE [25], and NSJADE [26], three state-of-the-art MOEAs that have been widely used for solving low to medium-dimensional robust multi-objective optimization problems. NSGA-II, NSCDE, and NSJADE are all dominance-based MOEAs, the difference of which lies in the selection of search engine. In NSGA-II, a real-coded genetic algorithm is utilized as the search engine; for NSCDE and NSJADE, two advanced DE variants CoDE and JADE are playing the role of the search engine.

The IGD and HV values of CNSDE/DVC, NSGA-II, NSCDE, and NSJADE are calculated and given in Table III. The box plots for the IGD and HV values after 30 runs are also provided in Fig. 5. In addition, the median attainment surfaces of 30 independent runs of these four MOEAs are shown in Fig. 6 [32]. It can be observed that CNSDE/DVC performs the best among the four MOEAs. Although the search engine of CNSDE/DVC is merely a simple original DE when compared with that of NSGA-II, NSCDE, and NSJADE, CNSDE/DVC performs the best among the four MOEAs. This is because the DVC operation decomposes the high-dimensional robust

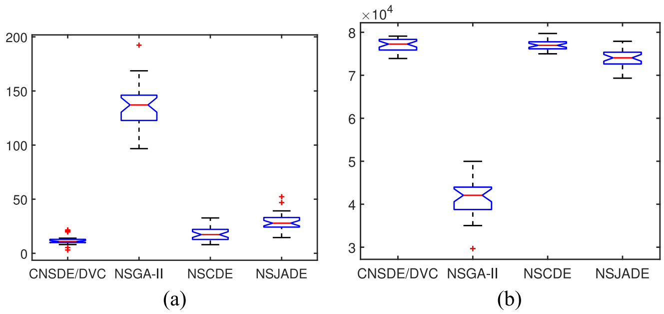


Fig. 5. Box plots for the IGD and HV values of CNSDE/DVC, NSGA-II, NSCDE, and NSJADE after 30 runs. (a) IGD. (b) HV.

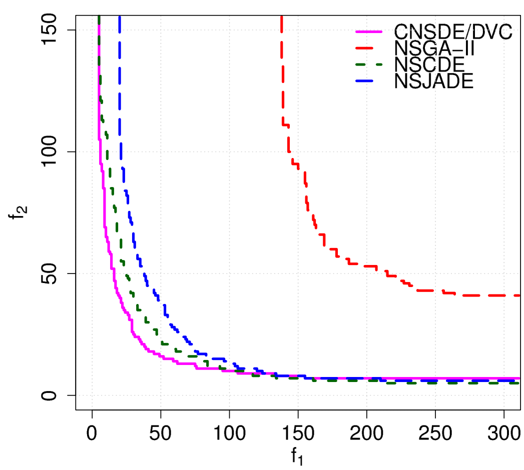


Fig. 6. Median attainment surfaces of 30 independent runs of CNSDE/DVC, NSGA-II, NSCDE, and NSJADE.

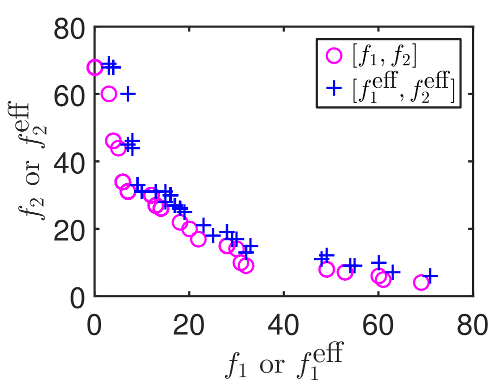


Fig. 7. Comparison of $[f_1, f_2]$ and $[f_1^{\rm eff}, f_2^{\rm eff}]$ obtained by CNSDE/DVC.

optimization problem, which reduces the complexity of the problem.

D. Analysis of Robust Order Schedules

We take a closer look at the robust solutions obtained by CNSDE/DVC. As introduced in Section II, f_1 and f_2 are two objectives, in the calculation of which the daily production quantities are fixed. While $f_1^{\rm eff}$ and $f_2^{\rm eff}$ are the mean effective objectives, which consider the impact of the uncertainty. In Fig. 7, we plot the values of $[f_1, f_2]$ and $[f_1^{\rm eff}, f_2^{\rm eff}]$ of the nondominated solutions sorted from CNSDE/DVC. From Fig. 7, it can

TABLE IV
DETAILED INFORMATION OF THE ORDER ASSIGNMENT IN THE SCHEDULE

Production Line No.	Order Assignment
1	$O_{18}(320)$, $O_{25}(400)$, $O_{12}(2921)$, $O_{15}(140)$, $O_{16}(200)$, $O_{19}(600)$, $O_{22}(800)$, $O_{23}(240)$, $O_{28}(480)$, $O_{30}(800)$, $O_{35}(696)$, $O_{32}(400)$, $O_{36}(800)$, $O_{37}(100)$
2	$O_{13}(174)$, $O_{17}(280)$, $O_{14}(270)$, $O_{24}(1000)$, $O_{15}(560)$, $O_{16}(300)$, $O_{19}(400)$, $O_{31}(260)$, $O_{33}(300)$, $O_{27}(200)$, $O_{38}(300)$, $O_{32}(600)$, $O_{36}(200)$
3	$O_2(600)$, $O_4(600)$, $O_9(800)$, $O_3(400)$, $O_{10}(800)$, $O_{18}(480)$, $O_{21}(320)$, $O_{26}(800)$, $O_{27}(800)$, $O_{35}(174)$, $O_{37}(400)$
4	$O_6(600)$, $O_9(200)$, $O_3(600)$, $O_{10}(200)$, $O_7(320)$, $O_5(200)$, $O_{11}(100)$, $O_{20}(850)$, $O_{34}(800)$, $O_{38}(200)$
5	$O_7(480)$, $O_8(320)$, $O_5(300)$, $O_{11}(400)$, $O_{14}(1079)$, $O_{12}(730)$, $O_{29}(800)$, $O_{22}(200)$, $O_{23}(160)$, $O_{28}(320)$, $O_{33}(200)$, $O_{39}(700)$
6	$O_1(3000)$, $O_2(400)$, $O_4(400)$, $O_6(400)$, $O_8(480)$, $O_{13}(696)$, $O_{21}(480)$, $O_{17}(420)$, $O_{25}(600)$, $O_{24}(250)$, $O_{29}(200)$, $O_{31}(390)$, $O_{40}(860)$

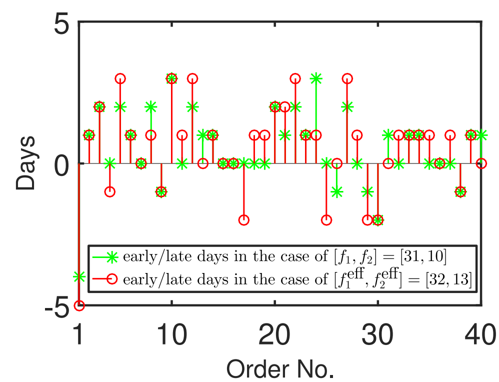


Fig. 8. Comparison of the early/late days of each order in the case of $[f_1, f_2] = [31, 10]$ and $[f_1^{\rm eff}, f_2^{\rm eff}] = [32, 13]$, respectively. (Negative values: early days; positive values: late days.)

be found that the uncertainty draws the PF away from the origin $[0, 0]$, which means the variations of daily production quantities greatly affect the total earliness and tardiness of all the orders.

A solution $[f_1, f_2] = [31, 10]$ (in the objective space) is chosen from the nondominated solutions in Fig. 7 for the analysis. It is also the knee point of the PF that balances the two optimization objectives. The details of the order assignment in the schedule represented by $[f_1, f_2] = [31, 10]$ are given in Table IV. The figures in the parentheses indicate the suborder size. When the uncertainty is taken into consideration, $[f_1^{\rm eff}, f_2^{\rm eff}] = [32, 13]$. In Fig. 8, we plot the early/late days of each order in the case of $[f_1, f_2] = [31, 10]$ and $[f_1^{\rm eff}, f_2^{\rm eff}] = [32, 13]$, respectively, where negative values denote early days and positive values represent late days. That the red circle overlaps the green asterisk indicates the early/late days of the order stay unchanged. It can be observed from Fig. 8 that the uncertainty has a major impact on the earliness and tardiness of the orders. There are total 23 of the 40 orders the early/late days of which are shifted. Robust order schedules can provide more accurate information on earliness/tardiness of orders, which helps planners pay close attention to the early/late orders. For the early orders, more warehouse spaces can be prepared as early as possible; for the late

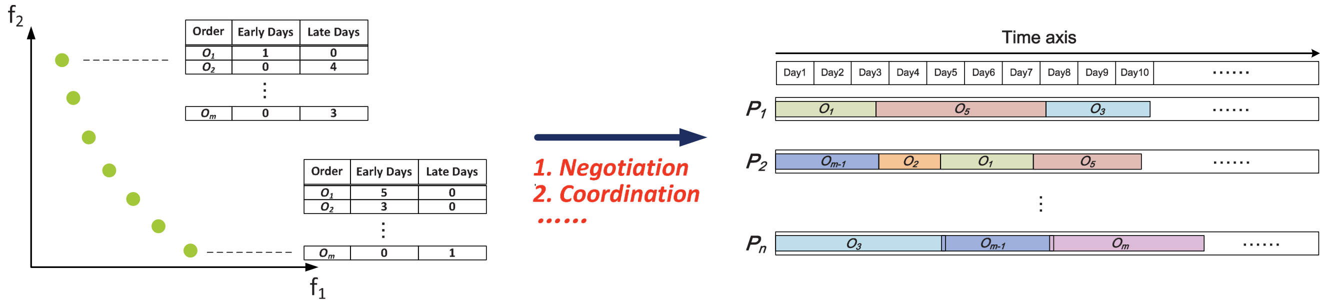


Fig. 9. Diagram that shows the final decision-making process.

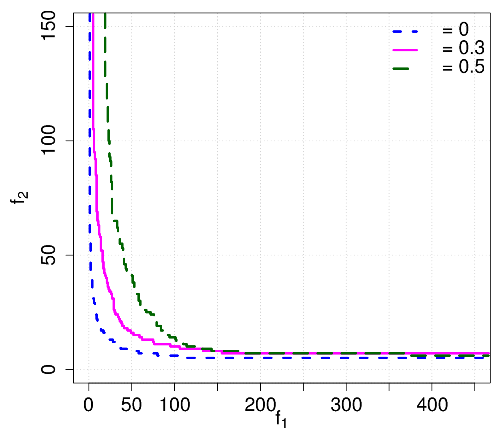


Fig. 10. Median attainment surfaces of 30 independent runs under different γ values.

orders, more operators can be arranged to work extra overtime on these orders.

Remark 3: It is noticed that a set of nondominated solutions are obtained by CNSDE/DVC. A decision-maker selects the proper solution (i.e., order schedule) from the Pareto front after negotiation and coordination. The negotiation and coordination are based on the early/late days of each order in the schedule. The diagram of the selection procedure is shown in Fig. 9. Each potential solution represents a possible schedule in which the early/late days of each order are determined. For potential late orders, the decision-maker needs to negotiate earlier with the customers who place the orders about the delay in delivery, or organize operators to work extra hours for these orders. For potential early orders, the decision-maker needs to arrange more warehouse spaces in advance. Therefore, according to the results of negotiation and coordination, the decision-maker can select the proper order schedule from the set of the nondominated solutions.

E. Effect of γ on Robust Order Scheduling

As introduced in Section II, γ indicates the amount of uncertainties considered in daily production quantities. Here, we discuss the effect of γ on robust order scheduling. γ is set as $[0, 0.3, 0.5]$, and the related median attainment surfaces of 30 independent runs under different γ values are shown in Fig. 10. It can be found that PFs gradually move away from the original point as γ increases. This phenomenon is predictable because larger γ brings about more uncertainties in the order scheduling, which makes the original nondominated solutions are no longer robust.

TABLE V
PERFORMANCE COMPARISON BETWEEN CNSDE AND CNSDE/DVC AFTER 30 RUNS IN THE CASE OF $D = 240$

Algorithm	IGD values (mean±std)	HV values (mean±std)
CNSDE	171.01±19.59†	9.93E+05±1.54E+04†
CNSDE/DVC	**43.09±16.58**	**1.10E+06±1.51E+04**

The better results are highlighted. "†" indicates that the result of the peer algorithm is significantly different from that of CNSDE/DVC at a 0.05 level by the Wilcoxon rank-sum test.

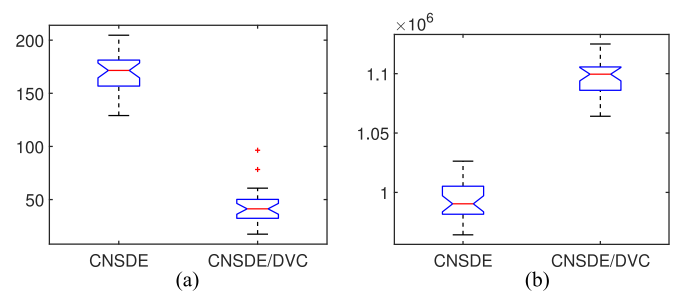


Fig. 11. Box plots for the IGD and HV values of CNSDE and CNSDE/DVC after 30 runs in the case of $D = 240$. (a) IGD. (b) HV.

F. Scalability Study

In the aforementioned experiments, 40 orders are considered and the dimension size of the problem is $D = 160$. Here, we consider more orders to test the scalability of CNSDE/DVC. Hence, 20 more orders are considered and the dimension size of the problem becomes $D = 240$. The details of these 20 orders are listed in Section S.VI of the supplementary file.

First, CNSDE/DVC is compared with CNSDE. The IGD and HV values are calculated after both algorithms are run for 30 times. The results are given in Table V. Additionally, the box plots for the IGD and HV values of CNSDE and CNSDE/DVC after 30 runs are provided in Fig. 11. Furthermore, the median attainment surfaces of 30 independent runs of CNSDE and CNSDE/DVC are provided in Fig. 12. It can be observed that CNSDE/DVC still works more efficiently than CNSDE when the dimension size of the problem increases.

Second, CNSDE/DVC is compared with NSGA-II, NSCDE, and NSJADE. IGD and HV values are calculated and given in

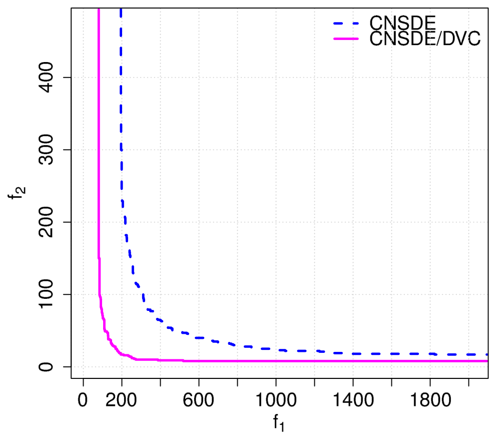


Fig. 12. Median attainment surfaces of 30 independent runs of CNSDE and CNSDE/DVC in the case of $D = 240$. (a) IGD. (b) HV.

TABLE VI
PERFORMANCE COMPARISON OF CNSDE/DVC, NSGA-II, NSCDE, AND NSJADE IN THE CASE OF $D = 240$

Algorithm	IGD values (mean±std)	HV values (mean±std)
CNSDE/DVC	**43.09±16.58**	**3.82E+05±1.00E+04**
NSGA-II	402.97±49.63†	1.60E+05±2.94E+04†
NSCDE	63.08±15.26†	3.78E+05±7.71E+03
NSJADE	70.26±21.90†	3.70E+05±1.21E+04†

The best results are highlighted. "†" indicates that the result of the peer algorithm is significantly different from that of CNSDE/DVC at a 0.05 level by the Wilcoxon rank-sum test.

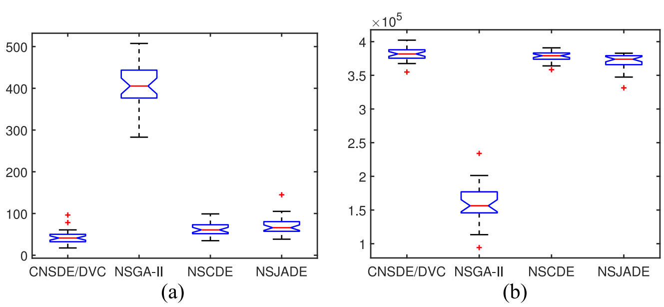


Fig. 13. Box plots for the IGD and HV values of CNSDE/DVC, NSGA-II, NSCDE, and NSJADE after 30 runs in the case of $D = 240$.

Table VI. The box plots for the related IGD and HV values after 30 runs are provided in Fig. 13. In addition, the median attainment surfaces of 30 independent runs of these four MOEAs are displayed in Fig. 14. It can be found that though the dimension size becomes larger, CNSDE/DVC still shows the best performance.

G. Computational Complexity Analysis

In this section, we discuss the computational complexities of the five MOEAs (i.e., CNSDE/DVC, CNSDE, NSGA-II, NSCDE, and NSJADE) in the experiments. In these five MOEAs, the common operations are nondominated sorting, crowding-distance assignment, and crowding-degree comparison. The worst-case complexities of the three operations are $\mathcal{O}(M(2NP)^2)$, $\mathcal{O}(M(2NP)\log(2NP))$, and $\mathcal{O}(2NP\log(2NP))$, respectively; the overall complexity is $\mathcal{O}(M(NP)^2)$ [13]. In addition, the search engines of the five MOEAs are the original DE or DE variants, the computational complexity is $\mathcal{O}(NP \cdot D)$. While for CNSDE/DVC, as shown in Algorithm 2, the computational complexity of DVC is $\mathcal{O}(TN \cdot SN \cdot D)$. Therefore, the computational complexities of CNSDE, NSGA-II, NSCDE, and NSJADE are $\mathcal{O}(M(NP)^2 + NP \cdot D)$; the computational complexity of CNSDE/DVC is $\mathcal{O}(M(NP)^2 + (NP + TN \cdot SN) \cdot D)$. The value of $TN \cdot SN$ is smaller than *NP*, which is often set as 100 in the experiment. Hence, the values of $NP \cdot D$ and $(NP + TN \cdot SN) \cdot D$ are largely determined by the value of D when the problem is high-dimensional. It can be found that the DVC operation does not substantially increase the computational cost of the original algorithm.

The above-mentioned experiments were carried out on a PC with Intel Core i7 Processor 3.60 GHz CPU and 4 GB RAM. The runtime of CNSDE/DVC for the robust order scheduling problem of 40 orders is around 1 hour when MAX_FES is set as $D \cdot 10\,000$. It is worth mentioning that order scheduling is made before the production, which can be regarded as an offline scheduling. Additionally, if high-performance computers and parallel computing are introduced to make the schedules in the factory, the scheduling time will further reduce. Meanwhile, intelligent order scheduling requires less manpower and fewer resources, which also saves the cost and increases the efficiency.

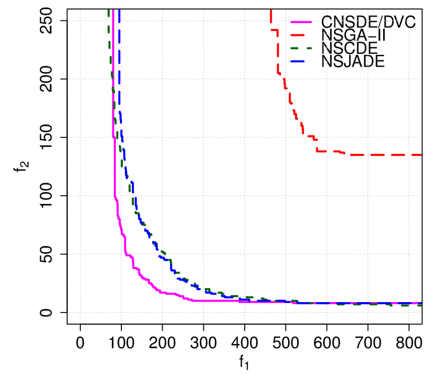


Fig. 14. Median attainment surfaces of 30 independent runs of CNSDE/DVC, NSGA-II, NSCDE and NSJADE in the case of $D = 240$.

V. CONCLUSION

This paper proposes a novel MOEA called CNSDE/DVC for solving high-dimensional robust multi-objective optimization problems with application to robust order scheduling. The high-dimensional decision variables are classified into highly robustness-related variables and weakly robustness-related variables based on their contributions to the robustness of candidate solutions; the two groups of decision variables are then optimized separately.

A group of numerical experiments have been conducted in the experimental section. The results reveal that the DVC-based approach is more efficient to solve high-dimensional robust order scheduling problems compared with three existing MOEAs developed for solving robust multi-objective optimization problems. The decision variables can be appropriately divided according to the their properties. It is worth mentioning that robust order schedules are able to provide more information on earliness/tardiness of orders. More warehouse spaces can be prepared as early as possible for the early orders, while more operators can be arranged to work overtime on the late orders.

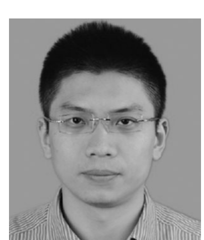
In the future, we are going to design a set of high-dimensional test functions, since there are no dedicated test functions for high-dimensional robust multi-objective optimization. Then, the performance of CNSDE/DVC will be examined on these test functions and other real-world high-dimensional robust optimization problems. In addition, we will design an adaptive parameter mechanism in CNSDE/DVC. We will also investigate whether more classifications of the decision variables can be made besides highly or weakly robustness-related variables.

Acknowledgment

The authors would like to thank the Associate Editor and the anonymous reviewers for their valuable comments and suggestions on this paper. They would also like to thank Dr. T. Chugh for his helpful suggestion in our revision.

References

[1] A. Şen, "The US fashion industry: A supply chain review," *Int. J. Production Econ.*, vol. 114, no. 2, pp. 571–593, 2008.
[2] H. K. Chan and F. T. Chan, "Early order completion contract approach to minimize the impact of demand uncertainty on supply chains," *IEEE Trans. Ind. Informat.*, vol. 2, no. 1, pp. 48–58, Feb. 2006.
[3] J. Branke, S. Nguyen, C. W. Pickardt, and M. Zhang, "Automated design of production scheduling heuristics: A review," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 110–124, Feb. 2016.
[4] J. R. Ashby and R. Uzsoy, "Scheduling and order release in a single-stage production system," *J. Manuf. Syst.*, vol. 14, no. 4, pp. 290–306, 1995.
[5] Z.-L. Chen and G. Pundoor, "Order assignment and scheduling in a supply chain," *Oper. Res.*, vol. 54, no. 3, pp. 555–572, 2006.
[6] Z. Guo, W. K. Wong, Z. Li, and P. Ren, "Modeling and Pareto optimization of multi-objective order scheduling problems in production planning," *Comput. Ind. Eng.*, vol. 64, no. 4, pp. 972–986, 2013.
[7] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *J. Scheduling*, vol. 12, no. 4, pp. 417–431, 2009.
[8] C. Og *et al.*, "Order acceptance and scheduling decisions in make-to-order systems," *Int. J. Production Econ.*, vol. 125, no. 1, pp. 200–211, 2010.
[9] W.-K. Wong, Z. Guo, and S. Leung, "Intelligent multi-objective decision-making model with RFID technology for production planning," *Int. J. Production Econ.*, vol. 147, pp. 647–658, 2014.
[10] Z. Guo, E. Ngai, C. Yang, and X. Liang, "An RFID-based intelligent decision support system architecture for production monitoring and scheduling in a distributed manufacturing environment," *Int. J. Production Econ.*, vol. 159, pp. 16–28, 2015.
[11] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, pp. 476–482, 2015.
[12] W. Du, S. Y. S. Leung, Y. Tang, and A. V. Vasilakos, "Differential evolution with event-triggered impulsive control," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 244–257, Jan. 2017.
[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[14] P. Chitra, R. Rajaram, and P. Venkatesh, "Application and comparison of hybrid evolutionary multiobjective optimization algorithms for solving task scheduling problem on heterogeneous systems," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2725–2734, 2011.
[15] Z. Guo, W. K. Wong, and S. Leung, "A hybrid intelligent model for order allocation planning in make-to-order manufacturing," *Appl. Soft Comput.*, vol. 13, no. 3, pp. 1376–1390, 2013.
[16] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.
[17] K. Deb and H. Gupta, "Introducing robustness in multi-objective optimization," *Evol. Comput.*, vol. 14, no. 4, pp. 463–494, 2006.
[18] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC2010 special session and competition on large-scale global optimization," Nature Inspired Comput. Appl. Lab., USTC, Hefei, China, Tech. Rep., 2009.
[19] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton, NJ, USA: Princeton Univ. Press, 2015.
[20] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 1994, pp. 249–257.
[21] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008.
[22] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 378–393, 2014.
[23] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 97–112, Feb. 2018.
[24] X. Ma *et al.*, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275–298, Apr. 2016.
[25] Y. Tang, H. Gao, W. Du, J. Lu, A. Vasilakos, and J. Kurths, "Robust multiobjective controllability of complex neuronal networks," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 13, no. 4, pp. 778–791, Jul./Aug. 2016.
[26] W. Du, Y. Tang, S. Y. S. Leung, L. Tong, A. V. Vasilakos, and F. Qian, "Robust order scheduling in the discrete manufacturing industry: A multiobjective optimization approach," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 253–264, Jan. 2018.
[27] Fast React, 2015. http://www.fastreact.com/
[28] Z. Guo, W. K. Wong, S. Leung, J. Fan, and S. Chan, "Mathematical model and genetic optimization for the job shop scheduling problem in a mixed-and multi-product assembly environment: a case study based on the apparel industry," *Comput. Ind. Eng.*, vol. 50, no. 3, pp. 202–219, 2006.
[29] K. R. Baker, "Minimizing earliness and tardiness costs in stochastic scheduling," *Eur. J. Oper. Res.*, vol. 236, no. 2, pp. 445–452, 2014.
[30] Y. Tang, H. Gao, J. Kurths, and J.-A. Fang, "Evolutionary pinning control and its application in UAV coordination," *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 828–838, Nov. 2012.
[31] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.
[32] M. López-Ibánez, L. Paquete, and T. Stützle, "Exploratory analysis of stochastic local search algorithms in biobjective optimization," in *Experimental Methods for the Analysis of Optimization Algorithms*. Berlin, Germany: Springer, 2010, pp. 209–222.



Wei Du (M'16) received the B.S. and M.S. degrees in electrical engineering from Donghua University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree in systems engineering from The Hong Kong Polytechnic University, Hong Kong, in 2016.

He is currently a Lecturer with the East China University of Science and Technology, Shanghai, China. His current research interests include evolutionary computation, especially differential evolution, evolutionary multiobjective optimization, robust evolutionary multi-objective optimization, and their applications.

Weimin Zhong received the B.S. degree in industry automation and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 1998 and 2006, respectively.

From 2006 to 2008, he was a Postdoctoral Research Fellow with the East China University of Science and Technology, Shanghai, China. From September 2013 to August 2014, he was a Visiting Research Fellow with the Department of Chemical Engineering, Lehigh University, Hangzhou, China.

He is currently a Professor and the Vice Dean of the School of Information Science and Engineering, East China University of Science and Technology. His current research interests include modeling, control, optimization and integration of industrial process, evolutionary multiobjective optimization, machine learning, and their applications. He authored or coauthored more than 80 papers and hosted more than ten projects including National Science Foundation of China, "863" Project, National Key Technology R&D Program.

Dr. Zhong is a recipient of Natural Science Fund for Excellent Young Scholar.

Yang Tang (M'11) received the B.S. and Ph.D. degrees in electrical engineering from Donghua University, Shanghai, China, in 2006 and 2010, respectively.

From 2008 to 2010, he was a Research Associate with Hong Kong Polytechnic University, Hong Kong. From 2011 to 2015, he was a Postdoctoral Researcher with the Humboldt University of Berlin, Berlin, Germany, and with the Potsdam Institute for Climate Impact Research, Potsdam, Germany. Since 2015, he has been a Professor with the East China University of Science and Technology, Shanghai, China. He has authored more than 60 refereed papers in international journals. His current research interests include multiagent systems/complex networks, cyber-physical systems, hybrid dynamical systems, and artificial intelligence, and their applications.

Prof. Tang was a recipient of the Alexander von Humboldt Fellowship in and the ISI Highly Cited Researchers Award in Computer Science by Clarivate Analytics in 2017. He is an Associate Editor of Scientific Reports, the *Journal of the Franklin Institute, Neurocomputing*, *Proceedings of the Institution of Mechanical Engineers, Part I*, the *Journal of Systems and Control Engineering*, and the *International Journal of Control Automation, and Systems*, and a Leading Guest Editor of the *Journal of the Franklin Institute*.

Wenli Du received the B.Sc. and M.Sc. degrees in chemical process automation from Dalian University of Technology, Dalian, China, in 1997 and 2000, respectively, and the Ph.D. degree in control theory and control engineering from East China University of Science and Technology, Shanghai, China, in 2005.

She is currently the Dean of the School of Information Science and Engineering, and the Vice Dean of the Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education in China, Beijing, China. Her research interests include modeling, control, optimization and fault diagnosis of chemical process, especially on hybrid semiparametric modeling methods based on the fusion of mechanism and data-driven methods, and optimization methods dealing with multiobjective and dynamic optimization. The research results have been widely applied to industrial ethylene plants and PTA plants, which brings significant economic and social benefits for SINOPEC and PetroChina.

Yaochu Jin (M'98–SM'02–F'16) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.Ing. degree from Ruhr University Bochum, Bochum, Germany, in 2001.

He is currently a Professor with Computational Intelligence, Department of Computer Science, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He is also a Finland Distinguished Professor funded by the Finnish Agency for Innovation (Tekes) and a Changjiang Distinguished Visiting Professor appointed by the Ministry of Education, China. He has (co)authored more than 250 peer-reviewed journal and conference papers and been granted eight patents on evolutionary optimization. He has delivered 30 invited keynote speeches at international conferences. His main research interests include data-driven surrogate-assisted evolutionary optimization, evolutionary multiobjective optimization, evolutionary learning, interpretable and secure machine learning, and evolutionary developmental systems.

Dr Jin is the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Co-Editor-in-Chief of *Complex and Intelligent Systems*. He is an IEEE Distinguished Lecturer (2013–2015 and 2017–2019) and past Vice President for Technical Activities of the IEEE Computational Intelligence Society (2014–2015). He is the recipient of the 2018 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, the 2015 and 2017 IEEE Computational Intelligence Magazine Outstanding Paper Award, and the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology.