A Framework for High-Dimensional Robust Evolutionary Multi-Objective Optimization

Wei Du
Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education
East China University of Science and Technology
Shanghai, China
duwei0203@gmail.com

Le Tong
The College of Information, Mechanical and Electrical Engineering
Shanghai Normal University
Shanghai, China
tongle0328@gmail.com

Yang Tang
Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education
East China University of Science and Technology
Shanghai, China
tangtany@gmail.com

ABSTRACT

This paper proposes a framework for solving high-dimensional robust multi-objective optimization problems. A decision variable classification-based framework is developed to search for robust Pareto-optimal solutions. The decision variables are classified as highly and weakly robustness-related variables based on their contributions to the robustness of candidate solutions. In the case study, an order scheduling problem in the apparel industry is investigated via the proposed framework. 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 multi-objective optimization problem.

KEYWORDS

Evolutionary multi-objective optimization, robust multi-objective optimization, high-dimensional optimization problem, evolutionary algorithm

ACM Reference Format:
Wei Du, Le Tong, and Yang Tang. 2018. A Framework for High-Dimensional Robust Evolutionary Multi-Objective Optimization. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3205651.3208243

1 INTRODUCTION

Over the past two decades and more, evolutionary multi-objective optimization (EMO) and multi-objective evolutionary algorithms (MOEAs) have attracted much attention due to their promising performance in finding a group of Pareto-optimal solutions for multi-objective optimization problems (MOPs) in a single run [20, 24]. While in many real-world MOPs, a broad range of uncertainties should be taken into consideration. These uncertainties can be divided into four categories [13]: 1) the fitness function involves noise; 2) the decision variables suffer from disturbances or changes; 3) the fitness function is subject to approximation errors; 4) the fitness function varies with time. And many investigations have been conducted along with these four directions [8, 11, 12, 21]. In this paper, we focus on the second category and investigate robust EMO.

It is widely known that the task of EMO is to search for a set of Pareto-optimal solutions in terms of a decision-maker's demand. While in practical situations, the decision variables (i.e., Pareto-optimal solutions) may be affected by external disturbances after optimization. For instance, in order scheduling problems, daily production quantities vary in real-world production due to multiple kinds of disruptions including tool failure, machine breakdown, and operator illness, among others [7]; therefore, under these circumstances, it is likely that the so-called optimal solutions will become suboptimal. In this regard, it is desirable to find the Pareto-optimal solutions which are robust to external disturbances in real-world MOPs, and it is also the ultimate goal of robust EMO [10].

Robust EMO has been investigated from various aspects, including searching for robust solutions for real-world applications [2, 7], defining robust multi-objective solutions [5], and designing robustness measures [9]. In these studies, the dimension size of the problems or test functions is often lower than 100, or even lower than 10 [2, 5, 16, 17]. Nevertheless, in many practical applications, the dimension size of the problem is much higher. For example, for an order scheduling problem of 30 orders with the consideration of some real-world production factors (e.g. order split, learning effect, etc.), the scheduling turns into a high-dimensional problem with more than 100 decision variables [7]. In addition, when the dimension size of a robust MOP increases, the robust region of the problem is much harder to determine. This is known as the "curse of dimensionality" [3], which implies that the performance of robust optimization methods becomes worse as the dimensionality of the search space grows. Therefore, it is of paramount importance to investigate how to solve high-dimensional robust MOPs in an effective way.

To solve high-dimensional *non-robust* optimization problems, Cooperative Coevolution (CC) is mostly utilized [18]. 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 [22] and differential grouping [15]. Recently,

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208243

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 [14]. Some promising experimental results are reported in [14, 23]. Inspired by the variable property-based classification, high-dimensional *robust* optimization problems can be tackled by identifying whether a decision variable is related to the robustness of candidate solutions.

Based on the above discussion, in this research, high-dimensional robust MOPs are investigated by a dedicated decision variable classification-based framework. To be specific, the framework is developed to search for robust Pareto-optimal solutions. In the framework, decision variables of the high-dimensional problem are classified as highly robustness-related variables and weakly robustness-related variables in terms of their contributions to the robustness of candidate solutions; then two groups of decision variables are optimized separately. The contributions of this research can be summarized from two aspects: 1) to the best of the authors' knowledge, it is the first attempt to investigate high-dimensional robust EMO; 2) decision variables are divided based on their influence on the robustness of candidate solutions.

This paper is organized as follows. Section 2 provides the background information of robust multi-objective optimization and the motivation of this work. Section 3 introduces the details of the decision variable classification-based framework. A case study is carried out in Section 4. Finally, concluding remarks are given in Section 5.

2 BACKGROUND INFORMATION AND MOTIVATION

In this section, we first provide the background information of robust multi-objective optimization. Then we explain the motivation of our work.

2.1 Robust Multi-Objective Optimization

In [5], the authors introduced robustness in multi-objective optimization by means of optimizing the mean effective objective functions instead of optimizing the original objective function. Moreover, two types of multi-objective robust solutions were defined in [5]. In the following paragraphs, we will introduce the related definitions in detail.

Multi-Objective Robust Solution of Type I: A solution $\mathbf{x}^*$ is called a multi-objective robust solution of type I, if it is the global feasible Pareto-optimal solution to the following multi-objective minimization problem (defined with respect to a δ-neighborhood of a solution $\mathbf{x}$):

$$\text{minimize} \quad (f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), ..., f_M^{\text{eff}}(\mathbf{x})), \quad \mathbf{x} \in \Omega, \tag{1}$$

where $f_i^{\text{eff}}(\mathbf{x})$ is defined as follows:

$$f_i^{\text{eff}}(\mathbf{x}) = \frac{1}{|\mathcal{B}_\delta(\mathbf{x})|} \int_{\mathbf{y} \in \mathcal{B}_\delta(\mathbf{x})} f_i(\mathbf{y}) d\mathbf{y}, \tag{2}$$

$\mathcal{B}_\delta(\mathbf{x})$ is a δ-neighborhood of a solution $\mathbf{x}$, $|\mathcal{B}_\delta(\mathbf{x})|$ is the hypervolume of the neighborhood; Ω is the feasible decision space, $\mathbf{x} = [x_1, x_2, ..., x_D]^T$ is a decision vector, and D is the dimension size, representing the number of the decision variables involved in the problem; $f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), ..., f_M^{\text{eff}}(\mathbf{x})$ are M mean effective objective functions for optimization.

However, the definition of robustness of type I is somewhat impractical. The reason is that a practitioner would be interested to know the limiting change in function values for defining robustness. Hence, the second type of robustness is defined as:

Multi-Objective Robust Solution of Type II: A solution $\mathbf{x}^*$ is called a multi-objective robust solution of type II, if it is the global feasible Pareto-optimal solution to the following multi-objective minimization problem:

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_M(\mathbf{x})), \\ \text{s.t.} \quad & \|\mathbf{f}^{\text{eff}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| \leq \eta, \\ & \mathbf{x} \in \Omega, \end{aligned} \tag{3}$$

where $\mathbf{f}^{\text{eff}}(\mathbf{x}) = (f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), ..., f_M^{\text{eff}}(\mathbf{x}))$; $\|\cdot\|$ can be any suitable norm; η is a constant which controls the desired level of robustness and the value is predefined by the practitioners.

In this research, the second type of robustness is utilized because it is more practical. For the calculation of f_i^{eff} in the above definition, a practical way is to generate a finite set of H solutions in a randomly or structured manner, which are selected around a δ-neighborhood $\mathcal{B}_\delta(\mathbf{x})$ of a solution $\mathbf{x}$ in the decision space; then the value of the mean effective objective function f_i^{eff} can be estimated by averaging the function values of the H neighboring solutions.

2.2 Motivation of This Work

In the previous research of robust EMO, the decision variables are treated as a whole in the optimization. However, based on our empirical observation, in some problems, part of the decision variables are highly related to the robustness of potential solutions; while the rest are weakly related to the robustness of potential solutions. To illustrate such an observation, we consider the following bi-objective optimization problem:

$$\begin{aligned} \text{minimize} \quad & f_1 = x_1, \\ & f_2 = 1 - x_1 + \tfrac{1}{x_2 + 0.2} \\ \text{s.t.} \quad & x_1, x_2 \in (0, 1). \end{aligned} \tag{4}$$

Based on Eq. (4), Figure 1 shows the changes to the robustness by sampling one decision variable while fixing the other to 0.4 and 0.8. The robustness is represented by $h = \|\mathbf{f}^{\text{eff}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\|_1$, where L^1-norm is utilized. When calculating $\mathbf{f}^{\text{eff}}(\mathbf{x})$, 50 solutions are generated around 0.1-neighborhood of a solution $\mathbf{x}$. It can be found that the robustness is nearly unchanged when sampling x_1; while the robustness varies as x_2 is sampled. Therefore, x_1 can be regarded as the decision variable that is weakly related to the robustness; x_2 is the decision variable that is highly related to the robustness. It is more beneficial to optimize x_1 and x_2 separately when searching for the robust Pareto-optimal solutions.

3 THE PROPOSED FRAMEWORK

The proposed framework aims to first decompose the decision variables and then optimize them separately for high-dimensional robust MOPs. There are three main components in the framework: *DV_CLAS*, *DV1_OPT* and *DV2_OPT*. First, *DV_CLAS* divides the

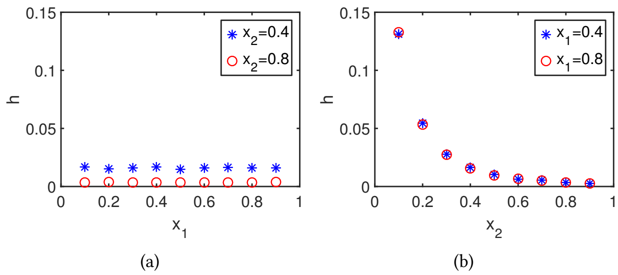

Figure 1: The changes to the robustness when sampling each decision variable. (a) Only sampling x_1 in the range of $(0, 1)$ when $x_2 = 0.4$ and $x_2 = 0.8$, respectively; (b) Only sampling x_2 in the range of $(0, 1)$ when $x_1 = 0.4$ and $x_1 = 0.8$, respectively.

Algorithm 1 The Proposed Framework

```
Begin
    /* NP: population size
    /* POP: current population
    /* NSe: number of selected individuals
    /* NSa: number of sampling on each decision variable of selected individuals
    /* DV1: highly robustness-related variables
    /* DV2: weakly robustness-related variables
    /* c1: number of cycle 1
    /* c2: number of cycle 2
    [DV1, DV2] = DV_CLAS(POP, NSe, NSa)
    while the maximum number of fitness evaluations is not achieved do
        for i1 = 1 : c1 do
            POP = DV1_OPT(POP, DV1)
        end for
        for i2 = 1 : c2 do
            POP = DV2_OPT(POP, DV2)
        end for
    end while
End
```

decision variables as two categories: highly robustness-related variables and weakly robustness-related variables. Then *DV1_OPT* and *DV2_OPT* optimize the two categories of variables for a certain number of cycles alternately until the stopping criterion is reached. In the following paragraphs, we will introduce these three components in detail.

3.1 *DV_CLAS*

As introduced in Section 2.1, the second type of robustness is used in this research, which means the robust optimization problem is converted into a constrained optimization problem. Thus searching for robust solutions is equivalent to searching for feasible solutions. Based on the above analysis, the main idea of *DV_CLAS* lies in sampling the decision variables, and then monitoring the changes to the constraint violation.

The details of the operation are given in Algorithm 2. Lines 10-15 describe sampling each decision variable of a number of individuals. First, *NSe* individuals are randomly selected from the current population. Then, *NSa* samplings are conducted on each decision variable of the *NSe* individuals, after which the variance values of the related constraint violation are recorded in *VarCV*.

Lines 16-20 present the classification based on the results from the sampling operation. First, we calculate for how many of the

Algorithm 2 *DV_CLAS*(*POP*, *NSe*, *NSa*)

```
1:  Begin
2:      /* V: dimension size of decision variables
3:      /* POP: current population
4:      /* NSe: number of selected individuals
5:      /* NSa: number of sampling on each decision variable of selected individuals
6:      /* DV1: highly robustness-related variables
7:      /* DV2: weakly robustness-related variables
8:      /* VarCV: size of NSe × V
9:      /* TVal: size of 1 × V
10:     for i = 1 : V do
11:         Randomly select NSe individuals from POP
12:         for j = 1 : NSe do
13:             Sample NSa times for the ith decision variable of the jth individual
    and record the variance of constraint violation values as VarCV_ji
14:         end for
15:     end for
16:     for i = 1 : V do
17:         TVal_i = sum(VarCV(:, i) == 0)
18:     end for
19:     Find the decision variables that meet TVal_i > mean_L(TVal) and record as
    DV2
20:     The rest decision variables are recorded as DV1
21: End
```

total *NSe* times for each decision variable the variance value of the constraint violation equals to 0. The values of the times are stored in *TVal*. Then the decision variables that meet $TVal > \text{mean}_L(TVal)$ are marked as weakly robustness-related variables, i.e., *DV2*; the rest decision variables are marked as highly robustness-related variables, i.e., *DV1*. It is worth mentioning that we use the Lehmer mean in the classification, which aims to select *DV2* with higher accuracy.

3.2 *DV1_OPT* and *DV2_OPT*

After grouping the decision variables into two categories, *DV1_OPT* and *DV2_OPT* are adopted to optimize each category of decision variables respectively. In both optimization strategies, any commonly used mutation/crossover operators (e.g., differential evolution [6], simulated binary crossover [1] and polynomial mutation [4]) can be utilized to generate the offspring population.

The difference between *DV1_OPT* and *DV2_OPT* lies in the selection operation. For *DV1*, they are highly related to robustness. We hope to obtain the solutions with high robustness by optimizing *DV1*. Therefore, robustness is used as the selection criterion for optimizing *DV1*, and individuals with higher robustness are preferred to survive in the next generation. While for *DV2*, they are weakly related to robustness. We aim to improve 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 higher nondomination rank and larger crowding distance will be chosen as the parents of the next generation.

In the optimization process, *DV1* and *DV2* are alternately optimized for c_1 and c_2 cycles respectively until the maximum number of fitness evaluations is reached.

4 CASE STUDY

4.1 Case Information

In this section, an order scheduling problem in the apparel industry is used as a case to examine the effectiveness of the proposed framework. In the apparel industry, the task of order scheduling is to appropriately assign n production orders to m lines for production. An illustration is provided in Figure 2.

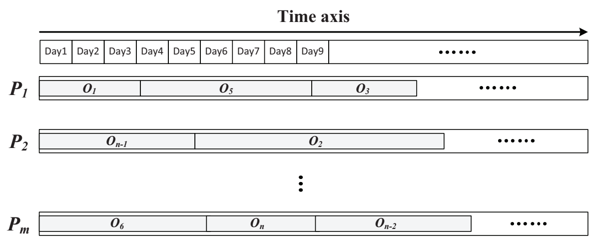

Figure 2: An illustration of order scheduling in the apparel industry. The order bar represents the duration of processing the related order.

An appropriate schedule implies that both earliness and tardiness of each order are discouraged. 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. As a result, the two optimization objectives of an order scheduling problem can be set as: 1) minimizing the total earliness of all the orders; 2) minimizing the total tardiness of all the orders.

In detail, the first objective is expressed as follows:

$$f_1 = \sum_{i=1}^{n} g_1(FD_i - DD_i), \tag{5}$$

where FD_i and DD_i are the finishing date and the due date of order i ($1 \leq i \leq n$) in the schedule, respectively; and $g_1(\cdot)$ is:

$$g_1(u) = \begin{cases} 0, & \text{if } u \geq 0, \\ -u, & \text{otherwise.} \end{cases} \tag{6}$$

The second objective is described as follows:

$$f_2 = \sum_{i=1}^{n} g_2(FD_i - DD_i), \tag{7}$$

where $g_2(\cdot)$ is:

$$g_2(u) = \begin{cases} 0, & \text{if } u \leq 0, \\ u, & \text{otherwise.} \end{cases} \tag{8}$$

These two objectives are usually 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 addition, during the production, orders can be split for flexible production [7]. Furthermore, 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 taken into account in the problem. The uncertainty of this problem comes from the uncertain daily production quantities, which affects FD_i of each order. As a result, we hope to obtain the schedules that are robust to the variations of daily production quantities.

In the encoding scheme used for this problem, there are three parts: the assignment of each order to the production line, the split percentage of each order, and the sequence of the orders on the same production line. Each order can be divided into at most two sub-orders; the split percentage is selected from $[0.2, 0.4, 0.6, 0.8]$. Therefore, the length of a potential solution is four times the number of the orders: $D = 4n$. Figure 3 illustrates the encoding scheme. When the number of the orders is more than 25, the problem becomes high-dimensional based on the encoding scheme.

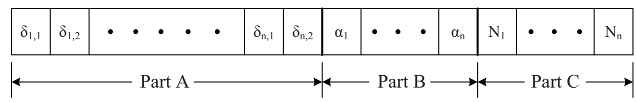

Figure 3: An illustration of the encoding scheme.

4.2 Experimental Setup

In the experiment, we consider 30 orders and 6 production lines. Hence the dimension size of the problem is $D = 120$. The maximum number of fitness evaluations (MAX_FES) is set as $D \cdot 10000$. The population size is $NP = 100$. We use a classical differential evolution (i.e., DE/rand/1/bin) [19] in the optimization. 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 $\beta = 0.3$. We set the number of the neighbouring points for each potential solution as $H = 5$. The desired level of robustness for this problem is predefined as $\eta = 5$.

In the proposed framework, we need to determine the values of NSa, NSe, c_1 and c_2. There are three parts in the encoding scheme of the problem. Therefore, according to the value range of each part, the settings are $NSa = 6$ for Part A, $NSa = 4$ for Part B, and $NSa = 30$ for Part C. The settings for the rest parameters are $NSe = 20$, $c_1 = 40$ and $c_2 = 8$.

We run each algorithm 15 times. Inverted generational distance (IGD) is used as the performance metric. To calculate IGD, a set of reference points need to be given beforehand. In this paper, the nondominated solutions obtained from the combined solutions of the algorithms in comparison are set as the reference points.

4.3 Effectiveness of the Proposed Framework

The main contribution of this research is to propose a framework to solve high-dimensional robust MOPs in a more efficient way. The core of the proposed framework is to decompose high-dimensional decision variables in terms of their influence on the robustness of candidate solutions. Therefore, here we examine the effectiveness of *DV_CLAS*. The algorithm with or without the classification operation is entitled FR or FR/noDVC respectively.

The nondominated solutions of the combined solutions obtained by FR/noDVC and FR after 15 runs are selected as the reference points. Then the IGD values of FR/noDVC and FR are calculated. For FR/noDVC, the mean and the standard deviation of the IGD is 35.74±3.44; while for FR, the value is 5.04±2.20. The Pareto fronts (PFs), i.e., robust order schedules, obtained from FR/noDVC and FR after 15 runs are also provided in Figure 4. It can be observed that FR greatly enhances the performance of FR/noDVC when handling the high-dimensional robust order scheduling problem.

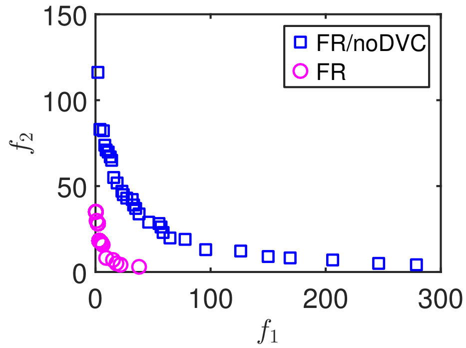

Figure 4: Comparison of the PFs (i.e., robust order schedules) obtained by FR/noDVC and FR after 15 runs.

In the proposed framework, the total 120 decision variables are classified as *DV1* (i.e., highly robustness-related variables) and *DV2* (i.e., weakly robustness-related variables). In Figure 5, we show the frequency of the decision variables that are labeled as *DV1* after 15 runs. It can be observed from Figure 5 that the decision variables from No. 61 to No. 90 are seldom identified as *DV1*. The reason can be inferred as follows: the 61st to the 90th decision variable represents the split percentage of each order. Compared to other decision variables, varying these 30 decision variables will only affect the sub-order size instead of the order sequence on each production line. Keeping the arrangement of the orders unchanged in a schedule (the sub-order size might be changed) indicates that the constraint violation keeps largely unchanged. Therefore, these 30 decision variables are most likely to be grouped into *DV2* instead of *DV1*.

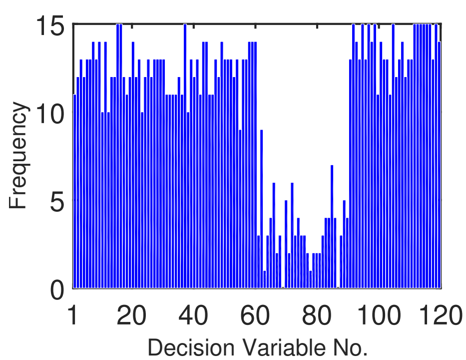

Figure 5: The frequency of the decision variables which are classified as *DV1* in the proposed framework after 15 runs.

4.4 Comparison with NSGA-II

To show the superiority of the proposed framework, we compare it with a popular MOEA: NSGA-II [5], which has been utilized for solving robust MOPs.

The nondominated solutions from the combined solutions obtained by FR and NSGA-II after 15 runs are used as the reference points. Then the IGD values of FR and NSGA-II are calculated. For FR, the mean and the standard deviation of the IGD is 5.04±2.19; while for NSGA-II, the value is 45.82±8.95. The PFs obtained from FR and NSGA-II after 15 runs are given in Figure 6. It can be observed that although the search engine of FR is merely a simple original DE when compared with that of NSGA-II, FR performs much better than NSGA-II. This is because the *DV_CLAS* operation decomposes the high-dimensional robust MOP, which reduces the complexity of the problem.

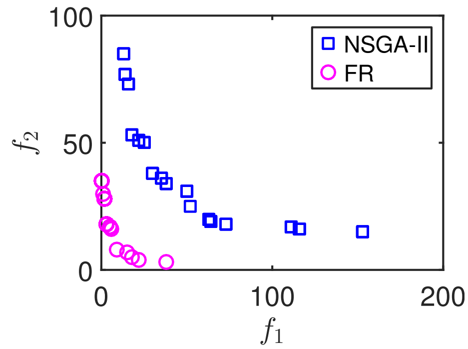

Figure 6: Comparison of the PFs (i.e., robust order schedules) obtained by FR and NSGA-II after 15 runs.

From the experimental results of the case study, it can be noticed that the proposed framework solves the high-dimensional robust MOPs effectively by classifying the decision variables and optimizing them separately.

5 CONCLUSION

In this paper, a novel framework is developed to solve high-dimensional robust MOPs. In the framework, high-dimensional decision variables are first classified as highly and weakly robustness-related variables based on their contributions to the robustness of candidate solutions; then the two categories of decision variables are optimized separately. In the case study, an order scheduling problem in the apparel industry is investigated by the presented framework. The experimental results reveal that the performance of robust EMO can be largely enhanced by the proposed framework.

In the future work, we hope to design a set of test functions for high-dimensional robust multi-objective optimization. In addition, we will focus on equipping the framework with parameter adaptation strategies.

ACKNOWLEDGEMENTS

This research was supported in part by the National Natural Science Foundation of China under Grant No. 61703163, in part by the China Postdoctoral Science Foundation under Grant No. 2016M601525, in part by the Fundamental Research Funds for the Central Universities under Grant No. 222201714028, and in part by Shanghai Sailing Program under Grant No. 17YF1427700.

REFERENCES

[1] Ram Bhusan Agrawal, Kalyanmoy Deb, and Ram Bhushan Agrawal. 1995. Simulated binary crossover for continuous search space. *Complex systems* 9, 2 (1995), 115–148.

[2] Md Asafuddoula, Hemant K Singh, and Tapabrata Ray. 2015. Six-sigma robust design optimization using a many-objective decomposition-based evolutionary algorithm. *IEEE Transactions on Evolutionary Computation* 19, 4 (2015), 490–507.

[3] Richard E Bellman. 2015. *Adaptive control processes: a guided tour.* Princeton university press.

[4] Kalyanmoy Deb and Mayank Goyal. 1996. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and informatics* 26 (1996), 30–45.

[5] Kalyanmoy Deb and Himanshu Gupta. 2006. Introducing robustness in multi-objective optimization. *Evolutionary Computation* 14, 4 (2006), 463–494.

[6] Wei Du, Sunney Yung Sun Leung, Yang Tang, and Athanasios V Vasilakos. 2017. Differential Evolution With Event-Triggered Impulsive Control. *IEEE Transactions on Cybernetics* 47, 1 (2017), 244–257.

[7] Wei Du, Yang Tang, Sunney Yung Sun Leung, Le Tong, Athanasios V Vasilakos, and Feng Qian. 2018. Robust Order Scheduling in the Discrete Manufacturing Industry: A Multiobjective Optimization Approach. *IEEE Transactions on Industrial Informatics* 14, 1 (2018), 253–264.

[8] Jonathan E Fieldsend and Richard M Everson. 2015. The rolling tide evolutionary algorithm: A multiobjective optimizer for noisy optimization problems. *IEEE Transactions on Evolutionary Computation* 19, 1 (2015), 103–117.

[9] António Gaspar-Cunha and José A Covas. 2008. Robustness in multi-objective optimization using evolutionary algorithms. *Computational Optimization and Applications* 39, 1 (2008), 75–96.

[10] Chi-Keong Goh and Kay Chen Tan. 2009. Robust Evolutionary Multi-objective Optimization. In *Evolutionary Multi-objective Optimization in Uncertain Environments*. Springer, 189–211.

[11] Shouyong Jiang and Shengxiang Yang. 2017. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 21, 1 (2017), 65–82.

[12] Yaochu Jin. 2005. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing* 9, 1 (2005), 3–12.

[13] Yaochu Jin and Jürgen Branke. 2005. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation* 9, 3 (2005), 303–317.

[14] Xiaoliang Ma, Fang Liu, Yutao Qi, Xiaodong Wang, Lingling Li, Licheng Jiao, Minglei Yin, and Maoguo Gong. 2016. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Transactions on Evolutionary Computation* 20, 2 (2016), 275–298.

[15] Mohammad Nabi Omidvar, Xiaodong Li, Yi Mei, and Xin Yao. 2014. Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation* 18, 3 (2014), 378–393.

[16] Yew-Soon Ong, Prasanth B Nair, and Kai Yew Lum. 2006. Max-min surrogate-assisted evolutionary algorithm for robust design. *IEEE Transactions on Evolutionary Computation* 10, 4 (2006), 392–404.

[17] Ingo Paenke, Jürgen Branke, and Yaochu Jin. 2006. Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Transactions on Evolutionary Computation* 10, 4 (2006), 405–420.

[18] Mitchell A Potter and Kenneth A De Jong. 1994. A cooperative coevolutionary approach to function optimization. In *International Conference on Parallel Problem Solving from Nature*. Springer, 249–257.

[19] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[20] David A Van Veldhuizen and Gary B Lamont. 2000. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation* 8, 2 (2000), 125–147.

[21] Handing Wang, Qingfu Zhang, Licheng Jiao, and Xin Yao. 2016. Regularity model for noisy multiobjective optimization. *IEEE Transactions on Cybernetics* 46, 9 (2016), 1997–2009.

[22] Zhenyu Yang, Ke Tang, and Xin Yao. 2008. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences* 178, 15 (2008), 2985–2999.

[23] Xingyi Zhang, Ye Tian, Ran Cheng, and Yaochu Jin. to be published, 2017. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Transactions on Evolutionary Computation* (to be published, 2017).

[24] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1, 1 (2011), 32–49.