


## Article

# Resilience-Based Surrogate Robustness Measure and Optimization Method for Robust Job-Shop Scheduling

Shichang Xiao <sup>1,\*</sup> , Zigao Wu <sup>2</sup> and Hongyan Dui <sup>3</sup><sup>1</sup> School of Logistics Engineering, Shanghai Maritime University, Shanghai 201306, China<sup>2</sup> Department of Mechanical Engineering, North China Electric Power University, Baoding 071003, China<sup>3</sup> School of Management Engineering, Zhengzhou University, Zhengzhou 450001, China

\* Correspondence: scxiao@shmtu.edu.cn

**Abstract:** This paper addresses the robust job-shop scheduling problems (RJSSP) with stochastic deteriorating processing times by considering the resilience of the production schedule. To deal with the disturbances caused by the processing time variations, the expected deviation between the realized makespan and the initial makespan is adopted to measure the robustness of a schedule. A surrogate model for robust scheduling is proposed, which can optimize both the schedule performance and robustness of RJSSP. Specifically, the computational burden of simulation is considered a deficiency for robustness evaluation under the disturbance of stochastic processing times. Therefore, a resilience-based surrogate robustness measure (SRM-R) is provided for the robustness estimation in the surrogate model. The proposed SRM-R considers the production resilience and can utilize the available information on stochastic deteriorating processing times and slack times in the schedule structure by analyzing the disturbance propagation of the correlated operations in the schedule. Finally, a multi-objective hybrid estimation of distribution algorithm is employed to obtain the Pareto optimal solutions of RJSSP. The simulation experiment results show that the presented SRM-R is effective and can provide the Pareto solutions with a lower computational burden. Furthermore, an RJSSP case derived from the manufacturing environment demonstrates that the proposed approach can generate satisfactory robust solutions with significantly improved computational efficiency.

**Keywords:** production resilience; robust job-shop scheduling; surrogate robustness measure; disturbance propagation; optimization algorithm

**MSC:** 90B36

**Citation:** Xiao, S.; Wu, Z.; Dui, H. Resilience-Based Surrogate Robustness Measure and Optimization Method for Robust Job-Shop Scheduling. *Mathematics* **2022**, *10*, 4048. <https://doi.org/10.3390/math10214048>

Academic Editor: Ripon Kumar Chakraborty

Received: 28 September 2022

Accepted: 26 October 2022

Published: 31 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Scheduling problems always play a key role in improving the production efficiency of a manufacturing system, such as flow shop scheduling [1,2], job-shop scheduling [3,4], and flexible job-shop scheduling [5–7]. The job shop is a typical workshop organization form to model the manufacturing environment. In recent years, tremendous research work has been conducted by considering a deterministic environment [8]. However, in a real-life job shop, the processing times would be deteriorated by many uncertainties, such as random machine breakdowns, uncertain setup times, and the unavailability of fixtures [9–11]. For example, in the aero-engine parts workshop, machine failures will cause processing interruption, and thus extend the processing time, or the tool wear will increase the actual processing time. Since the random deterioration of the processing time may decrease the production performance of a real-life manufacturing system, how to effectively deal with the job-shop scheduling problem (JSSP) with it has become a critical and interesting issue. There are four major approaches for dealing with such problems: a stochastic approach, a fuzzy approach, a stability approach, and a robust approach [12]. The stochastic approach [13] assumed that the job processing times are random variables with certain probability distributions

determined before scheduling. A fuzzy approach [14] allows a scheduler to find the best schedules, with respect to the fuzzy processing times of the jobs to be processed. The stability approach was developed to construct a minimal dominant set of schedules, which optimally covers all feasible scenarios. For example, Sotskov et al. [15] addressed a two-machine job-shop scheduling problem with fixed lower and upper bounds on the job processing times and provided a minimal dominant set of schedules to make the online scheduling decision. A robust approach considers the robustness to generate a schedule that can preserve a specified level of solution quality in the presence of uncertainties [16]. Generally, robustness has two definitions: quality robustness and solution robustness [17]. Quality robustness is used to indicate the insensitivity of the schedule performance under uncertainties, such as makespan or tardiness. The solution robustness refers to the insensitivity of activity start times to uncertainty [7,18]. In this study, we will focus on the quality robustness to deal with the robust job-shop scheduling problems.

For quality robustness, some studies used the minimized maximum regret value to evaluate it [8] and some regard the expected actual performance as the robustness [4,19]. For example, Wu et al. [20] proposed a decomposition method based on graph theory for JSSP, by using expected average weighted tardiness as a robustness measure. However, taking the expected performance as the robustness evaluation measure cannot evaluate the degree of influence of disturbances on the scheduling performance. To address this, the Pareto optimization method can be applied to optimize scheduling performance and robustness at the same time, just like when Xiong et al. [7] studied the flexible job-shop dual-objective robust scheduling problem with random machine failures. However, a new problem is that the robustness measure cannot be calculated analytically for the job-shop scheduling.

Robust scheduling based on simulation evaluation is an effective method [8], which can use the Monte Carlo simulation to simulate stochastic processing times to evaluate the robustness approximately. However, for a large number of processing time scenarios that are required to get a reasonable approximation, this type of method often bears an unacceptable computational burden [6,21]. By comparison, using the surrogate robustness measure (SRM) is an effective method to improve the computational efficiency of robustness evaluation [22,23]. In essence, the SRM is an approximate estimation measure that uses the partially known information of random disturbances and the structure of the schedule to decrease the computational burden of the robustness evaluation [13,24].

In recent years, researchers have begun to apply the surrogate model based on a surrogate measure to scheduling problems under the influence of uncertainties. Zahid et al. [25] investigated the surrogate measure of robustness for project scheduling problems. The authors stated that to avoid large tardiness due to disturbances, managers might opt for a more robust schedule rather than an optimal schedule. Therefore, they emphasized the importance of the SRMs that aid in identifying proactive schedules, which can provide flexible yet efficient plans with minimum redundancy. Wu et al. [26] address the optimization of risks in performance and stability (solution robustness) for the job-shop scheduling under random machine breakdowns. They considered three objectives: makespan, makespan risk, and stability risk. Then, they used the buffering approach under the limited predictive makespan to generate predictive schedules, which allows inserting additional idle time to control the risks. Zheng et al. [27] studied the production scheduling problem in an assembly manufacturing system with uncertain processing time and random machine breakdown. They considered the objectives of minimizing makespan and the performance deviation of the actual schedule from the baseline schedule. Xiong et al. [7] suggested two surrogate measures for the flexible job-shop scheduling problem with random machine breakdowns, considering both the available uncertainty information and the schedule structure. Yang et al. [28] studied a flexible job-shop schedule problem (FJSP) with random machine breakdown. They considered the makespan and robustness as the scheduling objectives. Then, they developed one surrogate measure named RMc to evaluate robustness using the heuristics of maximizing the workload and float time of each operation and the machine breakdowns.

From the above literature, although great efforts have been conducted for the robust scheduling problems in the field of production scheduling and project scheduling, a further study is urgently needed to consider the resilience of the manufacturing system to construct a more accurate surrogate robustness measure. System resilience is the ability of a system to respond and recover quickly from an external disruptive event [29]. In recent years, studies have been conducted considering system resilience. Mao et al. [30] formulates the supply chain networks problem as a network maximum-resilience decision, and thus developed the resilience of cumulative performance loss and the resilience of restoration rapidity to measure the resilience of the supply chain networks. Dui et al. [29] studied data-driven maintenance priority and resilience evaluation of performance loss in the main coolant system, in which the resilience of the MCS is represented by the loss and recovery of system performance. To effectively reduce and prevent losses, a resilience importance measure for performance loss is proposed to evaluate the performance of the main coolant system. Qi et al. [31] proposed a metro network's robustness and vulnerability measurement method under node interruption and edge failure based on the complex network modeling and topological characteristics analysis of metro systems. Robustness is an important connotation of system resilience; therefore, it has the potential to design better surrogate robustness measures based on resilience.

In the previous study, the authors made some contributions in the field of job-shop scheduling under uncertainties. The authors have considered the methods of robust scheduling and risk control to deal with the disturbances to improve the robustness and reduce the scheduling risk of the job shop. Xiao et al. [22] proposed an effective robustness surrogate measure for the job-shop scheduling problem with stochastic processing time and a hybrid estimation of the distribution algorithm. Wu et al. [23,24,26] studied the job-shop scheduling problem under random machine failures and proposed robust and/or stability optimization methods based on risk indicators. To further solve the JSSP under machine-worker dual resource constraints, Xiao et al. [32] proposed a multi-objective hybrid distribution estimation algorithm (MO-HEDA). At the same time, Dui et al. [29] carried out extended research in the field of system resilience and reliability. Although the existing research has made some progress, the robustness evaluation accuracy and the solution efficiency for solving the RJSSP are still restricted by uncertainties. This study focuses on the RJSSP under stochastic deteriorating processing times, which are assumed to follow the truncated normal distribution. The main contributions can be attributed as follows. First, we constructed a surrogate model of robust scheduling, considering the performance and the robustness at the same time, in which the robustness is evaluated by a surrogate robustness measure. Second, we proposed a new effective resilience-based surrogate robustness measure (SRM-R) that considers the disturbance propagation process. Moreover, we adopt a multi-objective hybrid estimation of the distribution algorithm (MO-HEDA) based on the research of Xiao et al. [18], which can provide the Pareto optimal solutions with a satisfactory performance by minimizing the SRM-R and the performance measure simultaneously.

The organization of this paper is as follows. Section 2 provides the problem description and the mathematical modeling of RJSSP. Section 3 investigates the existing SRMs and then presents an SRM based on the resilience and disturbance propagation among the operations. Section 4 provided the MO-HEDA for the dual-objective RJSSP. Section 5 verifies the effectiveness and performance of the presented SRM-R for bi-objective robust scheduling by three groups of computational experiments. Moreover, we show the effectiveness of the proposed surrogate model by using a scheduling problem instance derived from a real-life manufacturing workshop. Finally, we draw some conclusions and provide directions for future research in Section 6.

## 2. Problem Description and Surrogate Model

### 2.1. Problem Description

The notations that are used in the formulation of the RJSSP are defined in Table 1.

**Table 1.** The notations of the mathematical model.

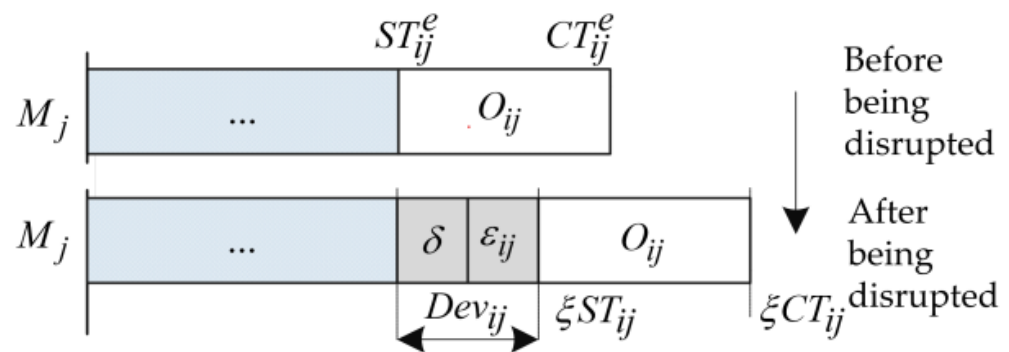
Notations	Definitions
$\xi$	The index of random variables
$n$	The total number of jobs
$m$	The total number of machines
$J_i$	The job $J_i$ , $i = 1, 2, \dots, n$
$M_j$	The machine $M_j$ , $j = 1, 2, \dots, m$
$O_{ij}$	The operation of the job $J_i$ processed on machine $M_j$
$\mu_{ij}$	The expected processing time of operation $O_{ij}$
$\sigma_{ij}^2$	The processing time variance of operation $O_{ij}$
$M_0$	A positive number that is large enough
$\xi C_{ij}$	The stochastic completion time of the job $J_i$ on machine $M_j$
$\xi T_{ij}$	The stochastic processing time of the job $J_i$ of on machine $M_j$
$C_{max}^e$	The makespan when the expected processing times are used
$a_{ijk}$	Job $J_i$ must be processed on a machine $M_j$ then on machine $M_k$ , if it is satisfied, $a_{ijk} = 1$ , otherwise, $a_{ijk} = 0$
$x_{ihj}$	Job $J_h$ must be processed after $J_i$ on machine $M_j$ , if it is satisfied, $x_{ihj} = 1$ , otherwise, $x_{ihj} = 0$

In the literature on stochastic scheduling, various types of probability distributions for stochastic processing times have been studied, such as exponential distribution, normal distribution, and uniform distribution. One method is to treat both the stochastic processing time and the scheduling objective as the stochastic variables, which aims at providing the processing ordering of the tasks on the machines with an expected optimization model. The historical processing data can be collected and used to get the probability distribution. To model the uncertain processing time, the actual job processing time is usually generated from a normal distribution with its expected value and the standard deviation, such as Gu et al. [19], Horng et al. [4], Wang et al. [13,33]. Therefore, the normal distribution is a representative type to study the stochastic processing times in the field of job-shop scheduling with uncertainty.

The RJSSP based on a classic job-shop scheduling problem is described as follows:  $n$  jobs processed on  $m$  machines; each job has  $m$  operations. The probability distribution of the stochastic processing times is derived from the historical manufacturing data. In this study, all the stochastic processing times,  $T_{ij} = \mu_{ij} + \varepsilon_{ij}$ ,  $\forall i, j$ , are assumed to follow the normal distribution, i.e.,  $T_{ij} \sim N(\mu_{ij}, \sigma_{ij}^2)$ , where  $\varepsilon_{ij}$  denotes the value of deteriorating disturbances on the expected processing time scenario. The processing information of any operation  $O_{ij}$  is denoted as a vector  $(M_j, \mu_{ij}, \sigma_{ij}^2)$ . Let  $\Omega$  denotes the feasible solution set that meets the process constraints. A feasible schedule is denoted by  $S$ .

In an arbitrary schedule, the expected processing time  $\mu_{ij}$  is a nominal value, then the total start time deviation of each operation is affected by two factors. The first factor is the internal disruption  $\varepsilon_{ij}$  that occurs in the operation  $O_{ij}$ . The second factor is the unabsorbed deviation caused by the preceding job  $J_{i'}$  on the same machine  $M_j$  or by the preceding machine  $M_{j'}$  of the job  $J_i$ . We use Figure 1 to explain how the deviation in operatn  $O_{ij}$  is caused.

Figure 1 shows the deviation of the start time of the operation  $O_{ij}$ , caused by the disruption of the machine  $M_j$ . Before the disruption occurs, the start time and completion time of  $O_{ij}$  are denoted by  $ST_{ij}^e$  and  $CT_{ij}^e$ , respectively. After the disruption occurs, the estimated start time and completion time are denoted by  $\xi ST_{ij}$  and  $\xi CT_{ij}$ , respectively. Thus, the total deviation of start time is indicated by  $Dev_{ij} = \delta + \varepsilon_{ij}$ , where  $\delta$  denotes the disruption caused by the preceding operation of the same job or the preceding job on the same machine.



**Figure 1.** The start time deviation after being disrupted.

## 2.2. Mathematical Formulation of the Surrogate Model

We use the Pareto optimization method to perform robust scheduling, thus, providing decision makers with a set of Pareto solutions that meet different risk preferences. The surrogate model of RJSSP is described as  $Jm|SPT|(f_1, f_2)$  by a triplet proposed by Graham et al. [34], where  $Jm$  denotes the JSSP,  $SPT$  denotes the stochastic deteriorating processing times, and  $f_1$  and  $f_2$  are the objectives that need to be optimized. Specifically, the SRM is used to evaluate the robustness of a schedule. The mathematical model of RJSSP is shown as follows:

$$\min_{S \in \Omega} \{f_1(S), f_2(S)\} \quad (1)$$

$$f_1(S) = C_{max}^e(S) = \max\{C_{ij}^e, \forall i, j\} \quad (2)$$

$$f_2(S) = SRM(S) \quad (3)$$

s.t.

$$\xi C_{ik} - \xi T_{ik} + M_0(1 - a_{ijk}) \geq \xi C_{ij} \quad (4)$$

$$i = 1, 2, \dots, n; j, k = 1, 2, \dots, m, j \neq k;$$

$$\xi C_{hj} - \xi T_{hj} + M_0(1 - a_{ihj}) \geq \xi C_{ij} \quad (5)$$

$$i, h = 1, 2, \dots, n, i \neq h; j = 1, 2, \dots, m;$$

$$\sum a_{ijk} = 1, \forall i, j, k, j \neq k; \quad (6)$$

$$\sum x_{ihk} = 1, \forall i, j, k, i \neq h; \quad (7)$$

$$a_{ijk} \in \{0, 1\}, \forall i, j, k, j \neq k; \quad (8)$$

$$x_{ihk} \in \{0, 1\}, \forall i, h, j, i \neq h; \quad (9)$$

$$\xi C_{ij} \geq 0, \forall i, h, j, i \neq h. \quad (10)$$

The objective functions are shown in Equation (1). The predictive makespan of a schedule is obtained by using the expected values of the processing times by function  $f_1(S)$ . Function  $f_2(S)$  is used to evaluate the robustness of a schedule solution. Inequality (4) represents the process constraints, which means the job  $J_i$  must be processed on the machine before being processed on the machine  $M_k$ .  $M_0$  is a large enough positive number. The machine constraints are denoted by inequality (5), which means the job  $J_h$  must be processed on the machine  $M_j$  after job  $J_i$ . Constraints (6) and (7) denote one machine can only process one job at one time and one job can only be processed on one machine at the same time. Constraints (8) and (9) represent the value of and can only be the integers 1 or 0. Inequality (10) denotes that none of the completion times of the operations can be negative.



### 3. The SRM-R for the Surrogate Model

Robust scheduling aims to generate the initial schedule with a satisfactory makespan and be capable of absorbing the disturbances caused by the deteriorating processing times when a right-shift reaction policy is adopted [35]. Generally, the robustness is estimated through Monte Carlo simulation by generating a certain number of processing times scenarios, which can be calculated by the following function:

$$RMsim(S) = \frac{1}{L} \sum_{l=1}^L [C_{max}^l(S) - C_{max}^e(S)] \quad (11)$$

where  $l = 1, 2, \dots, L$  denotes the number of simulation replications,  $RMsim$  denotes the simulation-based robustness,  $C_{max}^l(S)$  denotes the makespan of schedule  $S$  under the  $l$ th processing time scenario. In this paper, the processing time scenario indicates a group of potentially realized stochastic deteriorating processing times of all the operations.

The simulation-based robustness evaluation is time- and resource-intensive. Thus, the robustness of a schedule is usually evaluated by designing an SRM to avoid the computational burden caused by simulations. However, in the previous research, seldom effective SRM for RJSSP have been developed. We analyze five typical existing SRMs and then propose an SRM for RJSSP by using a statistical method and a disturbance propagation formula.

#### 3.1. The Existing Surrogate Robustness Measures

In the research of Leon et al. [35], an SRM for evaluating the absorption capability of a schedule was proposed based on the average total slacks. The measure is rewritten as SRM1, where  $TS_{ij}$  denotes the total slack time of the operation  $O_{ij}$ .

$$SRM1 = C_{max}^e(S) - \sum_{i=1}^n \sum_{j=1}^m \frac{TS_{ij}}{mn} \quad (12)$$

The SRMs that aim at providing an estimation of the schedule robustness for multi-mode project scheduling problems were investigated by Hazir et al. [36]. In one robustness measure, they define the activities that have the total slacks less than  $\xi$ ,  $\xi \in [0, 1]$  of the activity duration as the potentially critical activities (PCA), i.e.,  $PCA = \{O_{ij} : \frac{TS_{ij}}{\mu_{ij}} \leq \xi\}$ . The delays in PCA are likely to result in delays in the project completion time. Thus, those schedules with fewer critical activities are preferred on the robustness. The measure of PCA used in RJSSP is reasonable because of the similarity of the two problems, which is renamed potential critical operations (PCO). Considering the variance of RJSSP is assumed known, the definition of PCA is revised as the ratio of total slacks and the sum of planned processing time and standard deviation, i.e.,  $PCO = \{O_{ij} : \frac{TS_{ij}}{\mu_{ij} + \sigma_{ij}} \leq \xi\}$ . Therefore, the SRM is rewritten as follows:

$$SRM2 = \left\{ \frac{N_{PCO}}{mn}, O_{ij} \in \mathbf{O}_{PCO} \right\} \quad (13)$$

where  $N_{PCO}$  denotes the number of PCOs,  $\mathbf{O}_{PCO}$  denotes the set of PCO, which includes all the operations that satisfy the inequality  $\frac{TS_{ij}}{\mu_{ij} + \sigma_{ij}} \leq \xi$ .

Goren et al. [37] developed a surrogate stability measure to generate efficient and stable schedules for a job-shop subject to processing time variability and random machine breakdowns. The SRM is based on decreasing the maximum variance on the mean critical path, where the mean critical path is equivalent to the critical path of RJSSP when expected processing times are used. The SRM is written as follows:

$$SRM3 = \max \left\{ \sum VAR(O_{ij}), O_{ij} \in \mathbf{O}_{CP} \right\} \quad (14)$$

where the objective of SRM3 is to evaluate and select the critical path with the maximum sum variances, in which  $VAR$  denotes the variance of the processing times of the operations in the critical operation  $O_{CP}$ .

Xiao et al. [22] provided two SRMs for the robust scheduling of JSSP with stochastic processing times. They utilize the probability distribution information of stochastic processing times and the structure of the schedule solution determined by the slack times to estimate the robustness.

$$SRM4 = SM_{cp} + SM_{ncp} \quad (15)$$

In Equation (15),  $SM_{cp}$  and  $SM_{ncp}$  denote the estimated robustness degradation of the critical path and the non-critical path, respectively. The SRM4 aims to minimize the total robustness degradation of the baseline schedule.

$$SRM5 = \max\{SM_{cp}, SM_{ncp}\} \quad (16)$$

In Equation (16), SRM5 is provided to estimate the possible robustness degradation by minimizing the maximal robustness degradation on the critical path and non-critical path. In the following section, a disturbance propagation-based SRM is proposed, which is inspired by the disturbance absorption ability of a baseline schedule.

### 3.2. The Resilience-Based Surrogate Robustness Measures

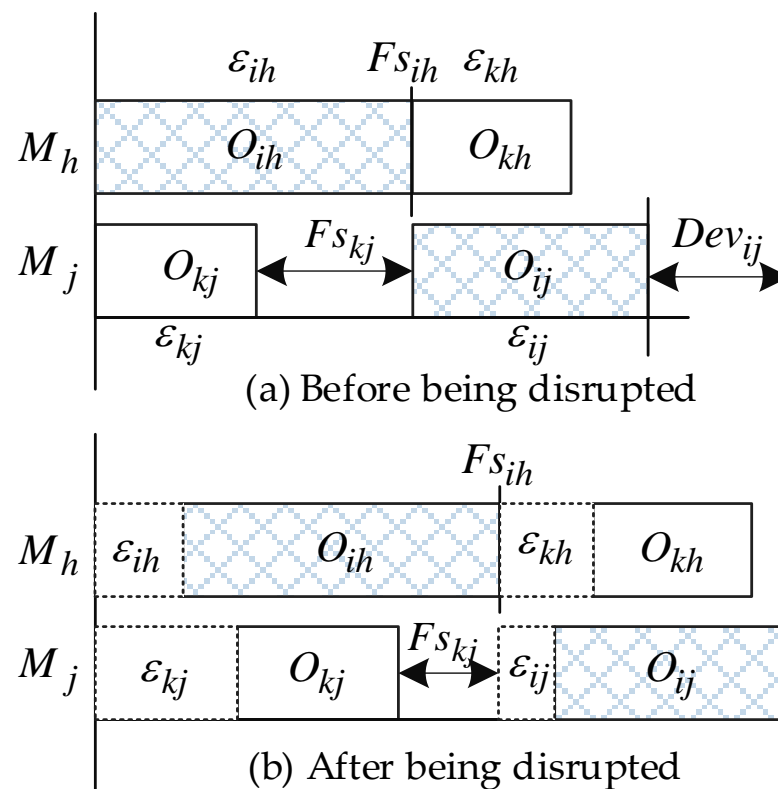
One of the core connotations of system resilience is the ability to absorb disturbances, which is reflected by the degree of system performance degradation under disturbance events. From this point of view, a schedule with good resilience may also have strong robustness. Therefore, we constructed a resilience-based surrogate robustness measure by analyzing the influence and propagation mechanism of stochastic deteriorating processing times to the makespan.

To describe the realized processing time scenario, the stochastic deteriorating disturbances were represented by  $\xi T_{ij} = \mu_{ij} + \varepsilon_{ij}$  when disruption was introduced. In this study, the processing times of each operation are considered to follow a normal distribution, i.e.,  $\xi T_{ij} \sim N(\mu_{ij}, \sigma_{ij}^2)$ . Then, map  $\xi T_{ij}$  into  $Z_{ij} = \frac{(\xi T_{ij} - \mu_{ij})}{\sigma_{ij}}$ , yielding  $Z_{ij} \in (0, 1)$ . Furthermore,  $1 - \alpha$  was denoted as the confidence level that is specified by the decision maker, considering the degree of risk aversion on the stochastic deteriorating processing times. We used the maximum possible value of the disruption with a certain confidence level to convert the disruption  $\varepsilon_{ij}$  into a confidence interval. Then, the worst case of the disruption was utilized for designing the SRM for robustness estimation. Since we are concerned about the upper limit of the realized processing time, the right-sided confidence interval of the normal distribution function was used. The probability is represented by Equation (17),

$$Pr\left\{Z_{ij} = \frac{(\xi T_{ij} - \mu_{ij})}{\sigma_{ij}} \leq Z_{\alpha}\right\} = 1 - \alpha, \quad (17)$$

where the value of realized processing time  $\xi T_{ij} \leq \mu_{ij} + Z_{\alpha}\sigma_{ij}$ ,  $Z_{\alpha}$  is the critical value determined by the predefined value of  $\alpha$ . Therefore, the possible disturbance of operation  $O_{ij}$  follows the equation  $\varepsilon_{ij} = \xi T_{ij} - \mu_{ij} \leq Z_{\alpha}\sigma_{ij}$ .

Since the disruptions among the operations propagate downstream under the process constraints, the recursion formula for calculating the  $Dev_{ij}$  of operation  $O_{ij}$  should be provided. Since the maximum value of the internal disruption  $\varepsilon_{ij}$  is set by using a predefined confidence level  $(1 - \alpha)$ , consideration is required for the worst case of all the processing time scenarios. The scenarios denote the realization of the stochastic processing time of the operations. The main difficulty, thereby, is to analyze the disturbances caused by the preceding operations. We use a problem instance with two consecutive jobs and two machines to analyze how to estimate the total deviation of an arbitrary operation, as shown in Figure 2.



**Figure 2.** An instance of two consecutive jobs processed on two machines.

We used Figure 2 to illustrate the schedule before and after the disruptions occur;  $\varepsilon$  denoted the possible maximum of the disruption of each operation;  $Fs_{kj}$  and  $Fs_{ih}$  denoted the free slacks of operation  $O_{kj}$  and  $O_{ih}$ , respectively, where  $Fs_{ih} = 0$ . Generally, the absorption ability of the schedule is determined by the free slacks that exist in the schedule. To estimate the robustness of a schedule, we analyzed the formula of each operation first and then extended the formula into a generic form. Furthermore, the generalized formula for estimating the deviation  $Dev_{ij}$  was used to develop the SRM for estimating the robustness of RJSSP.

The operation  $O_{kj}$  and  $O_{kh}$  are in the first position on the machine  $M_j$  and  $M_h$ . The deviation  $Dev_{kj}$  and  $Dev_{ih}$  are represented by  $Dev_{kj} = \varepsilon_{ij}$  and  $Dev_{ih} = \varepsilon_{ih}$ , respectively. The deviation of operation  $O_{ij}$  is possibly affected by its preceding operation  $O_{ih}$  or  $O_{kj}$  beside the disruption  $\varepsilon_{ij}$ . It was also found that there are some units of slack times before the operation  $O_{ij}$ , which can be regarded as the free slacks belonging to the operation  $O_{kj}$ . If  $Dev_{kj} = \varepsilon_{kj} > Fs_{kj}$ , the free slacks cannot absorb the disruption of operation  $O_{kj}$ , otherwise, the deviation of operation  $O_{kj}$  will not influence the starting time of operation  $O_{ij}$ . In addition, the deviation of operation  $O_{ih}$  may also affect the deviation of operation  $O_{ij}$ , since its free slack is  $Fs_{ih} = 0$ . Therefore, the total deviation  $Dev_{ij}$  was determined by the maximum deviation of its preceding job or operation. The available free slack is calculated by using the start time of  $O_{ij}$  and the maximum of the end time of operation  $O_{ih}$  and  $O_{kj}$ . From the above analysis, the  $Dev_{ij}$  is formulated as follows:

$$Dev_{ij} = \max\{[Dev_{ih} - (ST_{ij}^e - CT_{ih}^e)], [Dev_{kj} - (ST_{ij}^e - CT_{kj}^e)]\} + \varepsilon_{ij} \quad (18)$$

where  $ST^e$  and  $CT^e$  denoted the start time and completion time of the initial schedule, which are obtained by decoding the schedule using the nominal processing times  $\mu_{ij}$ . The reasonability lies in that the start times and completion times of the succeeding operation are shifted backward due to the adoption of the right-shift rescheduling method when disruption occurs. Therefore, the deviation  $Dev_{kh}$  of operation  $O_{kh}$  can be estimated by the following equation:



$$Dev_{kh} = \max\{[Dev_{kj} - (ST_{kh}^e - CT_{kj}^e)], [Dev_{ij} - (ST_{kh}^e - CT_{ih}^e)]\} + \varepsilon_{ij} \quad (19)$$

To generalize the formula for estimating the deviation on the arbitrary operation  $O_{ij}$ , we define  $O_{i'j}$  as the preceding operation of the operation  $O_{ij}$  on the same machine  $M_j$  and define  $O_{ij'}$  as the preceding operation of the same job  $J_i$ . There are two cases in which operation  $O_{ij}$  has no preceding operation or has no preceding job. The first case is that the operation  $O_{ij}$  is the first operation on a machine  $M_j$  while the second one is that  $O_{ij}$  is the first operation of the job  $J_i$ . To create a generic formula, four cases were discussed by considering whether the operation  $O_{ij}$  is the first operation on a machine  $M_j$  and whether operation  $O_{ij}$  is the first operation of the job  $J_i$ . We used  $M_j(p)|p = 1, \dots, n; j = 1, \dots, m$  to represent the  $p$ th job on the machine  $M_j$  and used  $J_i(q)|q = 1, \dots, m; i = 1, \dots, n$  to denote the  $q$ th operation of job  $J_i$ . Therefore, the deviation of the operation that considers the disturbance propagation was calculated by the following equation:

$$Dev_{ij} = \begin{cases} \varepsilon_{ij}, M_j(p=1) = O_{ij}, J_i(q=1) = O_{ij} \\ \max\{[Dev_{i'j} - (ST_{ij}^e - CT_{i'j}^e)], 0\} + \varepsilon_{ij}, M_j(p=1) = O_{ij}, J_i(q=1) \neq O_{ij} \\ \max\{[Dev_{ij'} - (ST_{ij}^e - CT_{ij'}^e)], 0\} + \varepsilon_{ij}, M_j(p=1) \neq O_{ij}, J_i(q=1) = O_{ij} \\ \max\{[Dev_{i'j} - (ST_{ij}^e - CT_{i'j}^e)], [Dev_{ij'} - (ST_{ij}^e - CT_{ij'}^e)], 0\} + \varepsilon_{ij}, \text{otherwise} \end{cases} \quad (20)$$

where  $M_j(p=1) = O_{ij}$  indicates that the first operation on the machine  $M_j$  is  $O_{ij}$  while  $J_i(q=1) = O_{ij}$  indicates that  $O_{ij}$  is the first operation of Job  $J_i$ . Due to the adoption of the railway execution strategy, the start time of  $O_{ij}$  cannot move forward, then the deviation of any operation cannot be negative. Moreover, the difference between the deviation and the free slack should be greater or equal to zero in the case that the disruption is fully absorbed. To further simplify the formula for estimating  $Dev_{ij}$ , we provide the following Proposition 1.

**Proposition 1.** For  $\forall O_{ij}, i = 1, \dots, n; j = 1, \dots, m$  in the schedule  $S \in \Omega$ , if the operation  $O_{ij}$  has at least one preceding job or one preceding operation, it holds that at least one of the two inequalities between  $Dev_{i'j} - (ST_{ij}^e - CT_{i'j}^e) \geq 0$  and  $Dev_{ij'} - (ST_{ij}^e - CT_{ij'}^e) \geq 0$  are satisfied in any case.

**Proof of Proposition 1.** For  $\forall O_{ij}, i = 1, \dots, n; j = 1, \dots, m$ , the following four cases cover all the possibilities:

Case 1, if  $M_j(p=1) = O_{ij}, J_i(q=1) = O_{ij}$ , there is no preceding operation and preceding job before the operation  $O_{ij}$ , it is easy to find that  $ST_{ij}^e = 0$ ;

Case 2, Else if  $M_j(p=1) = O_{ij}, J_i(q=1) \neq O_{ij}$ , there is at least one operation that precedes  $O_{ij}$  of job  $J_i$ , then  $ST_{ij}^e - CT_{ij'}^e = 0$ ;

Case 3, Else if  $M_j(p=1) \neq O_{ij}, J_i(q=1) = O_{ij}$ , there is at least one operation precedes  $O_{ij}$  on the machine  $M_j$ , then  $ST_{ij}^e - CT_{i'j}^e = 0$ ;

Case 4, Otherwise,  $M_j(p=1) \neq O_{ij}, J_i(q=1) \neq O_{ij}$ , there exists at least one operation that precedes  $O_{ij}$  of job  $J_i$  and at least one operation precedes  $O_{ij}$  on the machine  $M_j$ , then  $ST_{ij}^e - \max\{CT_{i'j}^e, CT_{ij'}^e\} = 0$ . It can be seen from the four cases that if an operation has no preceding jobs and preceding operations, its start time is equal to zero.

In other cases, there are at least one of the two equations of  $ST_{ij}^e - CT_{i'j}^e = 0$  and  $ST_{ij}^e - CT_{ij'}^e = 0$  holds. Since it also holds that  $Dev_{i'j} \geq 0$  and  $Dev_{ij'} \geq 0$ , therefore, at least one inequality in  $Dev_{i'j} - (ST_{ij}^e - CT_{i'j}^e) \geq 0$  and  $Dev_{ij'} - (ST_{ij}^e - CT_{ij'}^e) \geq 0$  is satisfied in any case.  $\square$

According to Proposition 1, the formula in Equation (20) is rewritten as Equation (21)

$$Dev_{ij} = \begin{cases} \varepsilon_{ij}, M_j(p=1) = O_{ij}, J_i(q=1) = O_{ij} \\ [Dev_{ij'} - (ST_{ij}^e - CT_{ij'}^e)] + \varepsilon_{ij}, M_j(p=1) = O_{ij}, J_i(q=1) \neq O_{ij} \\ [Dev_{ij'} - (ST_{ij}^e - CT_{ij}^e)] + \varepsilon_{ij}, M_j(p=1) \neq O_{ij}, J_i(q=1) = O_{ij} \\ \max\{[Dev_{ij'} - (ST_{ij}^e - CT_{ij'}^e)], [Dev_{ij} - (ST_{ij}^e - CT_{ij}^e)]\} + \varepsilon_{ij}, \text{otherwise} \end{cases} \quad (21)$$

By Equation (21), the completion time of each operation after being disrupted is estimated by Equation (22).

$$\zeta C_{ij} = ST_{ij}^e + Dev_{ij} + \mu_{ij} \quad (22)$$

According to Equation (22), the makespan is calculated by Equation (23).

$$\zeta C_{max} = \max \zeta C_{ij} = \max \{ST_{ij}^e + Dev_{ij} + \mu_{ij}\} \quad (23)$$

In Equation (23),  $ST_{ij}^e$ ,  $\mu_{ij}$  and the estimated disruption  $\varepsilon_{ij}$  in the formula of  $Dev_{ij}$  can be deemed as constants. Thereby, the only unknown parameter is the disturbance caused by the adjacent preceding operations and the available free slack times. For each schedule solution  $S \in \Omega$ , the estimated deviation of the makespan is calculated by Equation (24):

$$EDev(S) = \max \{ST_{ij}^e + Dev_{ij} + \mu_{ij}, \forall i, j\} - C_{max}^e(S) \quad (24)$$

where  $EDev(S)$  denotes the estimated deviation between the makespan after the disruptions and the initial makespan, considering the nominal processing times.

To keep the higher robustness of a schedule  $S$ , the  $EDev(S)$  should be minimized. Therefore, the presented SRM is written as Equation (25):

$$SRM - R = \min_{S \in \Omega} EDev(S) \quad (25)$$

where  $SRM-R$  denotes the proposed SRM. Therefore, the schedule with a higher disruption absorption capability, which keeps a lower deviation of the makespan, was selected.

#### 4. A Dual-Objective Optimization Algorithm

The estimation of distribution algorithm (EDA) is an evolutionary algorithm that explores the space of candidate solutions by using an explicit probabilistic model [38]. In this study, the procedures of HEDA presented by Xiao et al. [22,32,39] are adopted as the basis of MO-HEDA for solving the surrogate model of RJSSP.

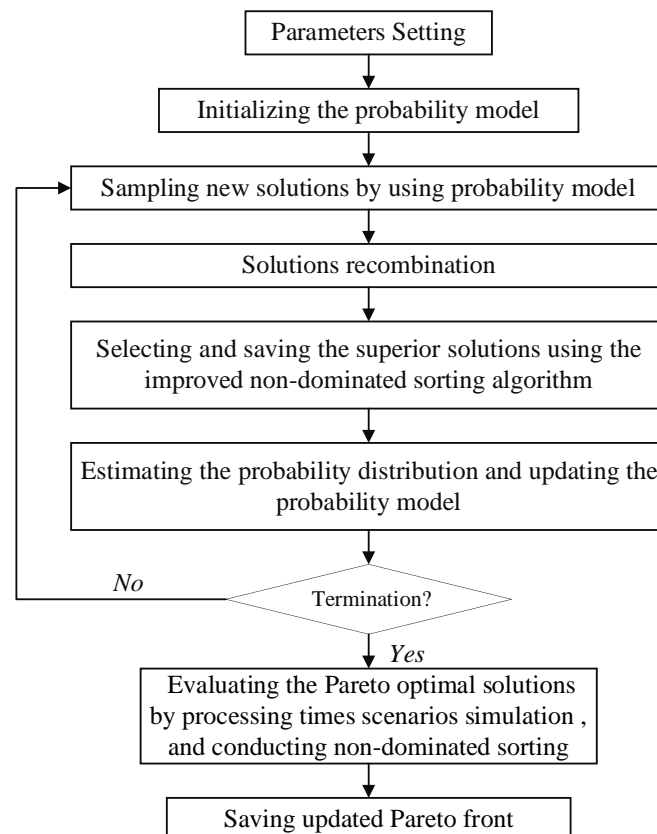
In this section, an improved multi-objective optimization algorithm based on the HEDA is adopted [32]. Before describing the MO-HEDA, the definitions are described:

**Definition 1.** *Pareto optimal solution: A schedule solution  $S^* \in \Omega$  is Pareto optimal solution if there is no other solution  $S \in \Omega$  that dominates  $S^*$ .*

**Definition 2.** *Pareto front: The Pareto front denotes the Pareto boundary constructed by the Pareto optimal solutions.*

**Definition 3.** *Dominance: For  $S_1, S_2 \in \Omega$ , there exist  $M$  objectives  $[f_1(S_i), \dots, f_k(S_i), \dots, f_M(S_i)]$  for each solution. It is said that  $S_1$  dominates  $S_2$ , if and only if  $i \in \{1, \dots, M\} : f_i(S_1) \leq f_i(S_2)$ , and there exists at least one  $h \in \{1, \dots, m\}, h \neq i$  satisfied  $f_h(S_1) \leq f_h(S_2)$ .*

The MO-HEDA is developed by combining the HEDA and NSGA-II [40,41]. The framework of the MO-HEDA is shown in Figure 3.



**Figure 3.** The framework of the MO-HEDA.

In the first three steps, the algorithm is used to get the Pareto optimal solution by running given generations. When RMSim is used as the robustness measure, simulation-based evaluation is adopted in the evolutionary process, then the Pareto front can be obtained directly. Then, the Pareto front will be further evaluated by Step 4 when SRM is used as the robustness measure. After being evaluated by simulation, the values of SRMs are turned into simulation-based robustness, which can be compared with the robustness obtained by simulation-based robustness. The pseudocode of MO-HEDA is shown in Algorithm 1.

---

**Algorithm 1.** The pseudocode of MO-HEDA.

---

Step 1. Set the parameters: population size  $P_{size}$ , the number of parents and offsprings  $\mu$ ,  $\lambda$ , the number of superior solutions  $B_{size}$ , evolutionary generation  $Iteration$ , recombination probability  $P_c$ , inheriting rate  $\theta$ , learning rate  $\alpha$ .

Step 2. Initialize the algorithm.

Step 2.1 Generate  $P_{size} = N$ , initial schedule solutions, let  $P_t = \{S_i | i = 1, \dots, N; t = 0\}$  denote the initial solution set,  $t$  is the index of iteration.

Step 2.2 Decode the solution to obtain the fitness value of each objective  $[f_1(S_i), f_2(S_i), \dots, f_M(S_i)]$ ;  $M$  denotes the number of objectives.

Step 2.3 Calculate the probability matrix  $P(0)$ .

Step 3. Set  $t = 1$

While  $t \leq Iteration$  Do

Step 3.1 1 Generate  $Pop_{size} = N$  new solutions by using the probability matrix  $P(t - 1)$ . Let  $Q_t = \{S_i | i = 1, \dots, N\}$  denote the newly generated solution set.

Step 3.2 Merge the solution set  $P_t$  and  $Q_t$  to get a new solution set  $P_t = P_t \cup Q_t$ .

Step 3.2.1 Execute the *improved non-dominated sort algorithm* to obtain the non-dominated levels of the Pareto solutions. Let  $F = \{F_1, F_2, \dots\}$  denote the solution set in all the non-dominated levels, where  $F_1$  denotes the first level.

---

---

Step 3.2.2 Set  $i = 1$   
 While  $P_t < N$  Do  
   If  $|P_t| > |P_t| + |F_i|$   
      $P_t = P_t \cup F_i$   
      $i = i + 1$   
   Else if  $|P_t| \leq |P_t| + |F_i|$   
      $P_t = P_t \cup \{\text{select } N - |P_t| \text{ solutions in } F_i\}$ , the solutions are selected by using the crowded-comparison operator;  
 End While  
 Step 3.2.3 Sort the solutions in the new solution set  $P_t$ .  
 Step 3.3 Select the first  $\mu = N$  solutions in  $P_t$  as the parent solutions.  
 Step 3.4 Generate  $\lambda = N$  offspring solutions by using the recombination method with the recombination probability  $P_c$  and then sort the  $(\mu + \lambda)$  solutions by using the same procedure as Step 3.2 to obtain an updated solution set  $P_t$ .  
 Step 3.5 Select  $B_{size}$  superior solution to update the probability matrix  $\mathbf{P}(t)$  of HEDA and save  $N$  elites in the solution set  $P_t$  or the next iteration by using the crowded-comparison operator.  
 $t = t + 1$ ;  
 End While  
 Step 4. Robustness evaluation by Monte Carlo simulation method.  
 Step 4.1 Conduct the simulation evaluation on the obtained Pareto optimal solutions in the solution set  $P_t$ .  
 Step 4.2 Sort the solutions in  $F_1$  according to the simulation results by using the *improved non-dominated sort algorithm*.  
 Step 4.3 Save the Pareto frontier of the simulated solutions.

---

In the MO-HEDA, we used an operation-based encoding scheme [22]. Each in the population denotes a scheduling solution of the RJSSP. The operations of the job  $J_i$  are represented by  $[(i-1)m+1, \dots, (i-1)m+j, \dots, (i-1)m+m]$ . Let  $\pi := (\pi_1, \dots, \pi_k, \dots, \pi_{n \times m})$ , where  $k := 1, 2, \dots, n \times m$  denotes the  $n \times m$  positions in each solution. Under the encoding scheme,  $(i-1)m+j$  in the position of  $\pi_k$  in  $\pi$  denotes that the  $j$ th operation of the job  $J_i$  is located at the  $k$ th position of  $\pi$ . The individual will never produce infeasible solutions in both the population initialization process and the individual sampling process, since it is encoded under precedence constraints.

The probability model is a vital procedure for the learning process of HEDA. The offspring solutions in each generation are produced by sampling the probability model. A probability matrix  $P(t) = \{p_{ij,\pi_k}^t | \forall i, j, k\}$  was defined considering the encoding scheme of RJSSP, where the element  $p_{ij,\pi_k}^t$  represents the probability of the operation on the machine  $M_j$  of a job  $J_i$  located in the position  $\pi_k$ . The probability function of the superior population is written as Equation (26):

$$p_{ij,\pi_k}^t = \sum_{s=1}^{B_{size}} \frac{I_{ij,\pi_k}^s(t)}{B_{size}}, \quad (26)$$

where  $B_{size}$  denotes the quantity of the superior population,  $I_{ij,\pi_k}^s(t)$  denotes the statistical function, according to Equation (27):

$$I_{ij,\pi_k}^s(t) = \begin{cases} 1, & \text{if operation } O_{ij} \text{ appears in position } \pi_k \text{ of the schedule} \\ 0, & \text{otherwise} \end{cases} \quad (27)$$

To update the probability matrix of each generation, an incremental learning rule in the field of machine learning was adopted. The learning function is shown in Equation (28).

$$p_{ij,\pi_k}^{t+1} = (1 - \alpha)p_{ij,\pi_k}^t + \alpha \frac{1}{B_{size}} \sum_{s=1}^{B_{size}} I_{ij,\pi_k}^s(t) \quad (28)$$

The recombination method of HEDA is designed considering the encoding scheme and the requirement of inheriting the excellent structure of the parent solutions [13,22]. With the recombination scheme, feasible solutions are generated directly and do not need a repairing process. As described by Xiao et al. [39], HEDA adopts an operation-based individual recombination method. This operator relies on the parent's operation inheritance rate,

which represents the number of jobs that need to be inherited from the relative position and is used to adjust the inheritance degree of the offspring to the parent's structural characteristics. To perform the recombination process, we first set the value of the parent process inheritance rate and the recombination probability; second, we selected  $\mu$  parent individuals and  $N$  positioning jobs, and found the positioning operations of jobs and the relative positions of the parent individuals, named P1 and P2, respectively. At last, we extracted all the operations of the remaining jobs, inserted the remaining operations of P1 into the vacancy of the offspring S2 in sequence, and inserted the remaining operations of P2 into the vacancy of the offspring S1 in sequence.

The non-dominated sort algorithm is an innovative algorithm for sorting a population into different non-domination levels. However, the algorithm can be further improved by reducing the storage space complexity of the solutions and the number of comparisons in the sorting process. The computational complexity of the non-dominated sort algorithm is  $O(MN^2)$ , where  $M$  denotes the number of objectives and  $N$  denotes the size of the population. As stated by Deb et al. [40], although the computational time complexity has reduced to  $O(MN^2)$ , the storage requirement has increased to  $O(N^2)$ . Therefore, there are two aspects of the previous algorithm that can be further improved. First, the comparison of each solution is conducted twice. Second, the solution set is dominated by the arbitrary solution  $S_i \in P_t, \forall i$  will be stored in the cache, while the dominated set will not be outputted. Therefore, the improved non-dominated sort algorithm is adopted. The procedures are shown in Algorithm 2.

---

**Algorithm 2.** The pseudocode of the improved non-dominated sort algorithm.

---

```

Step 1. Set the index of the solutions in the solution set  $P_t = \{S_i | i = 1, \dots, N\}$ , for each solution,
set  $n_i = 1$  as the number that is dominated.
Step 2. Determine the number that is dominated for each solution.
For  $i = 1, \dots, N - 1$ 
  For  $j = 1, \dots, N - i$ 
    If  $S_i$  dominates  $S_{i+j}$ 
       $n_{i+j} = n_{i+j} + 1$ ;
    Else if  $S_{i+j}$  dominates  $S_i$ 
       $n_i = n_i + 1$ ;
  End
End
Step 3. Sorting and storing the solutions in each front by using the times to be dominated.
Step 3.1 Sorting the solutions based on the  $n_i$  values of  $S_i \in P_t$  in ascending order.
Step 3.2 For  $i := 1 : N - 1$ 
  Set  $k := 1$ ,  $k$  denotes the number of solutions in each level.
  Set  $front = 1$ ,  $front$  is the index of the non-dominated level.
  If  $n_i = n_i + 1$ 
     $k := k + 1$ ;
     $F_{front} := F_{front} \cup S_{i+1}$ ;
  Else if  $n_i \neq n_i + 1$ 
     $front := front + 1$ ;
     $F_{front} := F_{front} \cup S_{i+1}$ ;
  End
End
Step 4. Return the solution sets  $F := \{F_{front} | front = 1, 2, \dots\}$ .

```

---

The overall complexity of the NSGA-II is  $O(MN^2)$ , which is governed by the non-dominated sorting algorithm. The improved non-dominated sorting algorithm reduces the storage space complexity from  $O(N^2)$  to  $O(N)$  by adopting the above well-designed algorithm structure. Meanwhile, the time complexity of comparisons is reduced to  $O(MN(N+1)/2)$  from  $O(MN^2)$  due to the solution comparison being reduced to once for each solution [32].

In the improved non-dominated sort algorithm, the niche selection technology is provided to maintain the diversity of Pareto solutions. In this way, the algorithm can converge to the uniformly distributed Pareto front. In this study, the crowding-distance assignment proposed by Deb et al. [40] is used. They defined density estimation to get an estimation of the density of solutions surrounding a particular solution in the solution set. The crowding distance  $i^{dis}$  is represented by the parameter of the cuboid formed by using the nearest neighbors as the vertices. For the arbitrary non-dominated solution set  $F_i$ , the crowding distance of each solution  $S_i$  can be calculated by the algorithm in Algorithm 3.

**Algorithm 3.** The crowding distance assignment.

---

**Step 1.** Let  $l = |F_{front}|$   
**Step 2.** For  $i = 1 : l$   
 Set  $S_{[i]distance} = 0$   
 For  $j = 1 : M$   
 $F_{front} = \text{sort}(F_{front}, j)$ , sort the  $j_{th}$  objective in an ascending order  
 Set  $S_{[1]distance} = S_{[l]distance} = \infty$   
 For  $i = 2 : l - 1$   
 $S_{[i]distance} = S_{[i]distance} + (S_{[i+1,j]distance} - S_{[i-1,j]distance}) / (f_j^{max} - f_j^{min})$   
 End  
 End  
 End  
**Step 3.** Save the  $S_{[i]distance}$  for the crowded-comparison operator.

---

In the algorithm of crowding distance assignment,  $S_{[i,j]dis}$  denotes the crowding distance of the  $j_{th}$  objective of the solution  $S_i$ ,  $f_i^{max}$  and  $f_i^{min}$  represents the maximum and minimum of the  $j_{th}$  objective in solution set  $F$ . By using the algorithm 2 and algorithm 3, we obtained two attributes of the solution, i.e., the non-domination rank  $n_i$  and crowding distance  $S_{[i]dis}$ .

## 5. Simulation and Analysis

### 5.1. Experiment Setting

The experimental analysis approach and the MO-HEDA are both coded by Matlab 2016 on a computer of Intel(R) Xeon(R) CPU E31230 @ 3.2 GHz with RAM of 8.00 GB, produced by Dell, Round Rock, Texas. The parameters of MO-HEDA set the same values for both RMs and SRM, which are listed in Table 2.

**Table 2.** The parameters setting for MO-HEDA.

Parameters of HEDA	Values
Population size	200
Evolutionary generations	200
Recombination probability	0.8
Number of positioning jobs	[0.5n]
Learning rate	0.3
Number of superior sub-population	40
Value of $\mu$ and $\lambda$	200
Number of simulation replications	50 or 200

The benchmark instances are modified from the classic JSSP benchmark instances by introducing the variance into the nominal processing times. The benchmark instances of FT06, FT10, and FT20 provided by Muth and Thompson [42] and LA06, LA16, LA21, LA26, and LA32 provided by Lawrence [43] are selected.



### 5.2. The Effectiveness Analysis of SRM-R

This section focuses on the correlation analysis between the SRMs and the RMs<sub>sim</sub> by using the coefficient of determination ( $R^2$  value). To analyze the  $R^2$  values under different degrees of uncertainty, we define the terminology of uncertainty level.

**Definition 4.** *Uncertainty level (UL): The uncertain level denotes the corresponding degrees of uncertainty in a RJSSP problem instance, which is measured by the percentage of uncertain operations relative to all the operations.*

The uncertain operations are selected randomly by a uniform distribution according to the specified UL. We choose five ULs (UL $x$ ,  $x = 2, 4, 6, 8, 10$ ) to compare the correlation between the SRMs and the corresponding simulation-based robustness (RMs<sub>sim</sub>). The value of RMs<sub>sim</sub> is also calculated using Equation (11).

The compared SRMs are optimized without considering the makespan. The critical value for possible maximal processing time disruption is set at  $Z_\alpha = 1.96$  with a confidence level of 97.5%. The  $R^2$  values are obtained from the regression of the values of SRM and the corresponding SRMs<sub>sim</sub> in the optimization process of HEDA with 200 generations. For each benchmark, five existing SRMs [22] and the presented SRM-R are tested. The average  $R^2$  values obtained by running HEDA ten times are shown in Table 3. Moreover, the average  $R^2$  values for all the selected ULs are provided to show the overall correlation of the comparison SRMs.

**Table 3.** The  $R^2$  value of the comparison SRMs.

Benchmark Instances	SRMs	Uncertainty Level					Average
		UL2	UL4	UL6	UL8	UL10	
FT06	SRM1	0.44	0.46	0.29	0.55	0.53	0.45
	SRM2	0.27	0.09	0.14	0.48	0.53	0.30
	SRM3	0.15	0.13	0.26	0.23	0.14	0.18
	SRM4	0.69	0.72	0.83	0.69	0.87	0.76
	SRM5	0.71	0.77	0.74	0.73	0.76	0.74
	SRM-R	0.84	0.81	0.78	0.82	0.78	0.81
FT10	SRM1	0.13	0.19	0.39	0.69	0.77	0.43
	SRM2	0.14	0.16	0.44	0.56	0.72	0.40
	SRM3	0.53	0.57	0.59	0.48	0.70	0.57
	SRM4	0.82	0.90	0.89	0.89	0.85	0.87
	SRM5	0.78	0.83	0.83	0.86	0.85	0.83
	SRM-R	0.86	0.92	0.96	0.89	0.91	0.91
FT20	SRM1	0.03	0.12	0.35	0.48	0.76	0.35
	SRM2	0.14	0.31	0.35	0.40	0.73	0.39
	SRM3	0.44	0.67	0.62	0.59	0.68	0.60
	SRM4	0.84	0.83	0.90	0.84	0.81	0.84
	SRM5	0.84	0.76	0.87	0.79	0.82	0.82
	SRM-R	0.92	0.96	0.94	0.96	0.92	0.94
LA06	SRM1	0.53	0.61	0.31	0.18	0.58	0.44
	SRM2	0.46	0.49	0.45	0.68	0.87	0.59
	SRM3	0.51	0.56	0.29	0.52	0.40	0.46
	SRM4	0.84	0.78	0.67	0.79	0.68	0.75
	SRM5	0.87	0.81	0.84	0.73	0.69	0.79
	SRM-R	0.92	0.93	0.93	0.97	0.98	0.95

Table 3. Cont.

Benchmark Instances	SRMs	Uncertainty Level					Average
		UL2	UL4	UL6	UL8	UL10	
LA16	SRM1	0.11	0.29	0.46	0.45	0.81	0.42
	SRM2	0.20	0.39	0.44	0.55	0.79	0.47
	SRM3	0.20	0.20	0.26	0.47	0.77	0.38
	SRM4	0.90	0.92	0.83	0.87	0.79	0.86
	SRM5	0.81	0.75	0.73	0.81	0.80	0.78
	SRM-R	0.92	0.89	0.93	0.92	0.92	0.92
LA21	SRM1	0.20	0.20	0.26	0.47	0.67	0.46
	SRM2	0.24	0.22	0.45	0.47	0.74	0.40
	SRM3	0.55	0.59	0.57	0.71	0.68	0.66
	SRM4	0.79	0.80	0.82	0.78	0.80	0.80
	SRM5	0.84	0.78	0.71	0.78	0.80	0.78
	SRM-R	0.90	0.88	0.92	0.90	0.88	0.90
LA26	SRM1	0.10	0.15	0.37	0.64	0.71	0.39
	SRM2	0.16	0.39	0.50	0.58	0.78	0.48
	SRM3	0.64	0.70	0.74	0.65	0.71	0.69
	SRM4	0.83	0.85	0.82	0.87	0.77	0.83
	SRM5	0.81	0.83	0.80	0.84	0.82	0.82
	SRM-R	0.90	0.92	0.94	0.91	0.94	0.92
LA32	SRM1	0.19	0.24	0.17	0.57	0.67	0.37
	SRM2	0.12	0.22	0.26	0.45	0.58	0.33
	SRM3	0.43	0.74	0.82	0.66	0.72	0.67
	SRM4	0.88	0.94	0.92	0.88	0.87	0.90
	SRM5	0.85	0.91	0.92	0.92	0.90	0.90
	SRM-R	0.92	0.93	0.94	0.91	0.93	0.93

### 5.3. The Performance Analysis of the Pareto Optimal Solutions

In this section, two different degrees of uncertainties are investigated by defining the benchmarks with 50% and 100% uncertain jobs as UL-M and UL-H, respectively, where UL-M denotes the medium uncertainty level and UL-H denotes the highest UL. To keep the benchmarks consistent when conducting robust scheduling by using SRM4 and RMsim, we select all the operations of the first  $[0.5n]$  jobs in each benchmark as the uncertain operations under UL-M, while all the operations of the  $n$  jobs will be selected as the uncertain operations under UL-H.

The simulation-based robustness measure is always regarded as the best one while with the largest computational burden [6,7]. Generally, the comparison is based on the set coverage between the Pareto front obtained by RMsim and that obtained by the proposed SRMs. In this study, we compare the performance of the presented SRM4 and RMsim directly, since the correlation values of the existing SRMs are apparently worse than those of the presented SRM4. To obtain the Pareto optimal solutions, the objective combinations of [makespan, RMsim] and [makespan, SRM-R] are used as the scheduling objectives. Two performance criteria are selected to conduct the performance comparison of RMsim and the presented SRM-R refers to the literature [7,44].

Set Coverage (SC): Let  $A$  and  $B$  be two Pareto solution sets, the set coverage  $SC(A, B)$  is defined as the percentage of the solutions in set  $B$  and is dominated by at least one solution in Pareto solution set  $A$ , as shown in Equation (29).

$$SC(A, B) = \frac{|\{x \in B | \exists y \in A : y \prec x\}|}{|B|} \quad (29)$$

A higher value of indicates a better performance of set  $A$ . Referring to the study of Xiong et al. (2013) [7], we use SC1 and SC2 to indicate the set coverage of the two non-dominated sets obtained by the robustness measures RMsim and SRM-R. For the

convenience of comparing the performance of SRM-R and RMsim, we define  $SC1 = SC(RMsim, SRM - R)$ ,  $SC2 = SC(SRM - R, RMsim)$ .

Computational time (CT): The computational time of running the MO-HEDA on the same benchmark instance by using RMsim and SRM-R.

Moreover, to show that a certain number of Pareto solutions can be obtained by using the proposed surrogate robustness measure, the number of Pareto optimal solutions is defined. Since RJSSP is an NP-hard problem and the exact optimal solution can hardly be obtained, the mentioned Pareto optimal solution is not the exact Pareto optimal solution. Given this, we use the optimal solutions obtained by MO-HEDA as the Pareto optimal solutions.

The number of Pareto optimal solutions (NPS): the number of Pareto optimal solutions on the Pareto front in the last generation.

For each benchmark, the solution performance obtained by SRM-R and RMsim will be compared using the above performance metrics; those with better performance are marked in bold. The RJSSP is a complex NP-hard combinatorial optimization problem. Moreover, the stochastic processing times also increase the randomness of the Pareto optimal solutions. Therefore, the MO-HEDA usually cannot get the same Pareto front over different runs of MO-HEDA. In the simulation, we find that the Pareto optimal solutions over different runs of MO-HEDA converge to nearly the same Pareto front under small-size benchmarks. However, under large-size benchmark instances, the Pareto optimal solutions are hard to converge to the same Pareto front due to the randomness of the optimal solutions of MO-HEDA and the effects of stochastic processing times.

From Table 4, we find that the average NPS obtained by RMsim is slightly greater than that of SRM-R under both UL-M and UL-H. The reason lies in that the SRM-R, which is an estimator of RMsim, would be slightly worse than RMsim. Moreover, the robustness evaluation in Step 4 of the MO-HEDA will also decrease the NPS in the first Pareto front. Meanwhile, the average, as well as the difference of the average under UL-H, are less than that under UL-M, which demonstrates under higher UL, the optimal Pareto solutions are harder to be obtained due to the variability of the objective values under uncertain processing times. When compared with the average NPS over all the benchmarks, the NPS of SRM-R is smaller than that of RMsim, although the difference between the average NPS values under UL-H is slight.

**Table 4.** The value of NPS.

UL	Benchmark Instances	FT06	FT10	FT20	LA06	LA11	LA16	LA21	LA26	LA32	Average
UL-M	RMsim	10.4	17.8	14.8	18.2	17.8	17	10	9.4	10.4	14.0
	SRM-R	7.2	8.4	10.8	11.4	10.2	10.2	8.6	9.4	7.8	9.3
UL-H	RMsim	6.0	10.6	9.3	8.5	8.5	7.1	9.3	7.4	6.0	8.1
	SRM-R	4.2	6.3	6.2	6.0	7.9	4.2	8.7	7.6	6.1	6.4

The performance of the Pareto optimal solutions obtained by SRM-R and RMsim is represented by set coverage, shown in Table 5. Under the UL-M for the nine benchmarks, the values of SC2 range from 0.66 to 0.98; the SRM-R shows better optimal performance in two benchmarks. The average values of SC1 and SC2 over all the benchmarks are 0.78 and 0.9, respectively. It means there is about a 12% performance deviation of the Pareto solutions obtained by SRM-R and those obtained by RMsim under UL-M from an overall perspective. As seen from the results under UL-H, there are four benchmarks that show the better performance of SRM-R, which is 44.4% of the total number of the tested benchmarks. Moreover, the average value of SC2 is 0.79, which is very close to the value of SC1, namely, 0.84. In this study, the direct comparison of the set coverage between RMsim and SRM-R shows that a higher SC2 is obtained, which is ranging from 0.62 to 1.0. This result is very satisfying when compared with the performance of the SRM proposed for flexible JSSP by Xiong et al. [7], which is rarely greater than 0.2 over the tested problems. Therefore, it is concluded that Pareto solutions can be obtained by using SRM-R as an estimator of robustness.

**Table 5.** The SC of the Pareto optimal solutions obtained by RMsim and SRM-R.

UL	Benchmark Instances	FT06	FT10	FT20	LA06	LA16	LA16	LA21	LA26	Average
UL-M	SC1	1.00	0.98	0.78	0.90	0.89	0.90	1.00	0.74	0.90
	SC2	0.66	0.67	0.92	0.75	0.78	0.85	0.74	0.98	0.78
UL-H	SC1	0.71	0.84	0.96	0.83	0.95	0.86	0.77	0.75	0.84
	SC2	0.95	0.72	0.73	0.62	0.66	0.89	0.97	1.00	0.79

The main advantage of the SRM-R is the computational efficiency. The percentage of the decreasing of the CT of SRM-R, compared with that of RMsim is calculated by Equation (30).

$$PD = 100 \times \frac{(CT_{RMsim} - CT_{SRMDP})}{CT_{RMsim}} \% \quad (30)$$

The average of the CT of ten replications, as well as the relative percentage deviation under both the UL-M and UL-H over the nine benchmarks are provided in Table 6.

**Table 6.** The comparison of average computational time (seconds).

UL	CT	FT06	FT10	FT20	LA06	LA16	LA16	LA21	LA26	Average
UL-M	SRM-R	71.31	137.94	146.30	110.44	137.62	183.15	261.58	448.63	71.31
	RMsim	317.10	546.58	499.50	382.56	527.04	680.27	844.29	1298.57	317.10
	PD (%)	77.51	74.76	70.71	71.13	73.89	73.08	69.02	65.45	77.51
UL-H	SRM-R	69.29	130.49	134.48	107.32	127.84	185.42	257.03	439.83	69.29
	RMsim	308.59	542.97	516.87	387.12	593.40	708.96	862.71	1325.46	308.59
	PD (%)	77.55	75.97	73.98	72.28	78.46	73.85	70.21	66.82	77.55

It is easily found that the CT of SRM-R is significantly lower than that of RMsim. The decreasing percentage of CT ranges from 65.45% to 77.51% under UL-M and ranges from 66.82% to 78.46% under UL-H. The results are obtained with simulation replications of 50.

According to the analysis, it is verified that the presented SRM-R can estimate robustness accurately. Therefore, the decision makers can obtain the Pareto optimal solutions using SRM-R as the robustness measure with a tremendously lower computational time. Meanwhile, the RMsim is preferred when there is enough simulation time or more Pareto solutions are needed, since it can provide more Pareto solutions. However, if the decision makers want to obtain the Pareto solutions with high computational efficiency, especially for large-size problem instances, the presented SRM-R is a better choice.

#### 5.4. Case Study

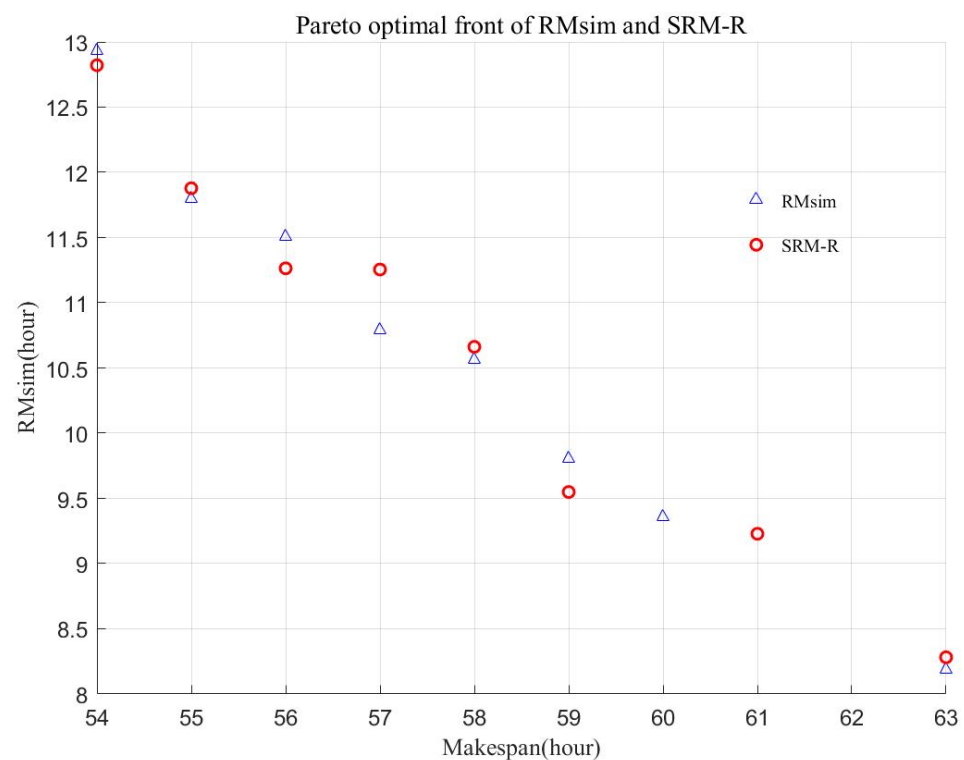
A typical problem instance derived from a job-shop manufacturing environment is selected [13]. The problem instance is extracted from the manufacturing enterprise and then modified to verify the effectiveness of the surrogate model for RJSSP. There are eight parts to be processed, which are typical aircraft engine parts. Each part requires six major processes, including turning, milling, planning, grinding, drilling, and boring. After the setup process of the machine tool is finished, it will then be machined by the machine tool automatically. The machine tools are numbered, respectively, including lathe (M1), milling machine (M2), planer (M3), grinding machine (M4), drilling machine (M5), and boring machine (M6). The sample mean and variance of the processing time of each process are obtained, according to the historical data. The processing times approximately follow the normal distribution, then the processing time scenarios are assumed to follow the trunked normal distribution. The corresponding machine information and processing time information (units/hour) are shown in the form of (the machine number, the expected value, and the variance) [13]. We solve the problem instance using the surrogate model and the simulation-based model of RJSSP, respectively. In each computational simulation, we use MO-HEDA to solve the problem instance 10 times, the Pareto front is selected to

compare the performance of the SRM-R and RMsim, shown in Table 7, the unit of Makespan and RMsim is the hour.

**Table 7.** The Pareto front obtained by SRM-R and RMsim.

Pareto Solutions	SRM-R		RMsim	
	Makespan	RMsim	Makespan	RMsim
1	57	11.25	60	9.36
2	54	12.82	58	10.56
3	56	11.26	56	11.50
4	58	10.66	57	10.79
5	59	9.57	59	9.82
6	61	9.22	54	12.93
7	55	11.90	63	8.18
8	63	8.28	55	11.79

The average computation time of SRM-R is 103.57 s. The average computation time of RMsim is 563.55 s. The decreasing percentage of the computation time is up to 81.62%. To further show the robust scheduling results of the SRM-R, the Pareto front of the SRM-R is provided in Figure 4.



**Figure 4.** The Pareto optimal front of the comparison robustness measures.

The variation range of RMsim is 8.18–12.93;  $(12.93 - 8.18) / 12.93 = 31.9\%$ . The range of Makespan is 54–64;  $(64 - 54) / 64 = 15.6\%$ , which means that the decision maker can choose any solution sacrificing at most 15.6% performance of the makespan while obtaining 31.9% improvements in the robustness. Comparing the Pareto optimal solutions obtained by SRM-R and RMsim in ten computational experiments, the set coverage value of the Pareto optimal fronts is 40% (RMsim dominates SRM-R, SC1) and 37.5% (SRM-R dominates RMsim, SC2), respectively. As for the Pareto front, the set coverage value is 50% (SC1) and 37.5% (SC2), respectively. The results denote that the SRM-R is effective to obtain robust schedule solutions with a significantly lower computational burden.

## 6. Conclusions

This paper addressed the RJSSP under stochastic deteriorating processing times, considering the resilience of production schedule, by formulating a surrogate model. To develop an effective SRM for robustness estimation, we provide a statistical approach that transforms the disturbances of stochastic deteriorating processing time to an estimated upper bound with a specified confidence level. Thereby, the formula of the disruption propagation is provided by analyzing the structure of the schedule. Additionally, then, we utilize the disruption propagation formula and the available processing time information to formulate the SRM-R. Moreover, we provide the MO-HEDA for solving the multi-objective RJSSP, in which the improved non-dominated sorting algorithm can increase computational efficiency by reducing the storage complexity and time complexity of the non-dominated sorting process.

Three groups of simulation experiments verified the effectiveness and the performance of the presented SRM-R. First, we investigated the correlation between the SRMs and the RMsim. The results demonstrated that the correlation of the presented SRM-R is higher than that of the existing SRMs, which indicates a better estimation accuracy of robustness. The second experiment compared the performance in obtaining the Pareto optimal solution obtained by using the presented SRM-R and the RMsim. According to the results, we concluded that the SRM-R can be used to estimate the robustness of RJSSP and obtain satisfying Pareto optimal solutions, with a tremendously lower computational burden. In addition, a case study further verified that the proposed surrogate model is effective for the studied RJSSP and production resilience can be effectively improved. Therefore, the presented SRM-R is a good choice for the decision makers, when computational efficiency is emphasized.

The limitation of the proposed SRM-R is the assumption that the processing time obeys the normal random separation. When the distribution type changes, the conditions for the disturbance propagation will change. Therefore, future research will further study the surrogate robustness measure under different probability distribution types, such as uniform distribution, lognormal distribution, and exponential distribution. Moreover, to further improve the production reliability, the surrogate measure of stability by using the disruption propagation formula should also be studied. Thereby, the multi-objective scheduling optimization approach for RJSSP will be provided, considering the makespan, the robustness, and the stability by using the presented surrogate measures. At the same time, this research may provide a theoretical basis for data-driven robust scheduling of job shops with the support of new technologies, such as digital twins and cloud manufacturing.

**Author Contributions:** Conceptualization, S.X. and Z.W.; methodology, S.X. and H.D.; software, S.X.; validation, S.X., Z.W. and H.D.; formal analysis, S.X.; investigation, S.X.; resources, Z.W.; data curation, Z.W.; writing—original draft preparation, S.X.; writing—review and editing, S.X., Z.W. and H.D.; visualization, S.X.; supervision, S.X. and H.D.; project administration, S.X.; funding acquisition, S.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Soft Science Key Project of Shanghai Science and Technology Innovation Action Plan (No. 20692193300); the National Natural Science Foundation of China (No. U1904211); the Fundamental Research Funds for the Central Universities (No. 9160421005); Shanghai University Young Teacher Training Grant Program: (No. ZZSH20010).

**Data Availability Statement:** The authors confirm that the data supporting the findings of this study are available within the article.

**Acknowledgments:** The authors are thankful for the valuable comments and suggestions provided by the anonymous reviewers on the drafts of our paper. We are also grateful to the Shanghai Frontiers Science Center of “Full penetration” Far-Reaching Offshore Ocean Energy and Power.

**Conflicts of Interest:** The authors declare no conflict of interest.



## References

1. Abdel-Basset, M.; Mohamed, R.; Abouhawwash, M.; Chang, V.; Askar, S. A Local Search-Based Generalized Normal Distribution Algorithm for Permutation Flow Shop Scheduling. *Appl. Sci.* **2021**, *11*, 4837. [\[CrossRef\]](#)
2. Liu, F.; Wang, S.; Hong, Y.; Yue, X. On the Robust and Stable Flowshop Scheduling Under Stochastic and Dynamic Disruptions. In *IEEE Transactions on Engineering Management*; IEEE: Manhattan, NY, USA, 2017; pp. 1–15.
3. Zhang, R.; Wu, C. An Artificial Bee Colony Algorithm for the Job Shop Scheduling Problem with Random Processing Times. *Entropy* **2011**, *13*, 1708–1729. [\[CrossRef\]](#)
4. Horng, S.-C.; Lin, S.-S.; Yang, F.-Y. Evolutionary algorithm for stochastic job shop scheduling with random processing time. *Expert Syst. Appl.* **2012**, *39*, 3603–3610. [\[CrossRef\]](#)
5. Wang, S.; Wang, L.; Xu, Y.; Liu, M. An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *Int. J. Prod. Res.* **2013**, *51*, 3778–3793. [\[CrossRef\]](#)
6. Al-Hinai, N.; ElMekkawy, T.Y. Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *Int. J. Prod. Econ.* **2011**, *132*, 279–291. [\[CrossRef\]](#)
7. Xiong, J.; Xing, L.-N.; Chen, Y.-W. Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. *Int. J. Prod. Econ.* **2013**, *141*, 112–126. [\[CrossRef\]](#)
8. Xiong, H.; Shi, S.; Ren, D.; Hu, J. A survey of job shop scheduling problem: The types and models. *Comput. Oper. Res.* **2022**, *142*, 105731. [\[CrossRef\]](#)
9. Aytug, H.; Lawley, M.A.; McKay, K.; Mohan, S.; Uzsoy, R. Executing production schedules in the face of uncertainties: A review and some future directions. *Eur. J. Oper. Res.* **2005**, *161*, 86–110. [\[CrossRef\]](#)
10. Ouelhadj, D.; Petrovic, S. A survey of dynamic scheduling in manufacturing systems. *J. Sched.* **2009**, *12*, 417–431. [\[CrossRef\]](#)
11. Sabuncuoglu, I.; Goren, S. Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *Int. J. Comput. Integr. Manuf.* **2009**, *22*, 138–157. [\[CrossRef\]](#)
12. Sotskov, Y.N.; Werner, F. *Sequencing and Scheduling with Inaccurate Data*; Nova Science Publishers, Inc.: New York, NY, USA, 2014.
13. Xiao, S.; Wu, Z.; Yu, S. A two-stage assignment strategy for the robust scheduling of dual-resource constrained stochastic job shop scheduling problems. *IFAC-PapersOnLine* **2019**, *52*, 421–426. [\[CrossRef\]](#)
14. Palacios, J.J.; González-Rodríguez, I.; Vela, C.R.; Puente, J. Robust multiobjective optimisation for fuzzy job shop problems. *Appl. Soft Comput.* **2016**, *56*, 604–616. [\[CrossRef\]](#)
15. Sotskov, Y.N.; Matsveichuk, N.M.; Hatsura, V.D. Two-Machine Job-Shop Scheduling Problem to Minimize the Makespan with Uncertain Job Durations. *Algorithms* **2019**, *13*, 4. [\[CrossRef\]](#)
16. Zhang, J.; Yang, J.; Zhou, Y. Robust scheduling for multi-objective flexible job-shop problems with flexible workdays. *Eng. Optim.* **2016**, *48*, 1973–1989. [\[CrossRef\]](#)
17. Herroelen, W.; Leus, R. Robust and reactive project scheduling: A review and classification of procedures. *Int. J. Prod. Res.* **2004**, *42*, 1599–1620. [\[CrossRef\]](#)
18. Goren, S.; Sabuncuoglu, I. Optimization of schedule robustness and stability under random machine breakdowns and processing time variability. *IEE Trans.* **2009**, *42*, 203–220. [\[CrossRef\]](#)
19. Gu, J.; Gu, M.; Cao, C.; Gu, X. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. *Comput. Oper. Res.* **2010**, *37*, 927–937. [\[CrossRef\]](#)
20. Wu, S.D.; Byeon, E.-S.; Storer, R.H. A Graph-Theoretic Decomposition of the Job Shop Scheduling Problem to Achieve Scheduling Robustness. *Oper. Res.* **1999**, *47*, 113–124. [\[CrossRef\]](#)
21. Xiao, S.; Sun, S.; Yang, H. Proactive Scheduling Research on Job Shop with Stochastically Controllable Processing Times. *J. Northwest. Polytech. Univ.* **2014**, *6*, 019.
22. Xiao, S.; Sun, S.; Jin, J. Surrogate Measures for the Robust Scheduling of Stochastic Job Shop Scheduling Problems. *Energies* **2017**, *10*, 543. [\[CrossRef\]](#)
23. Wu, Z.; Sun, S.; Xiao, S. Risk measure of job shop scheduling with random machine breakdowns. *Comput. Oper. Res.* **2018**, *99*, 1–12. [\[CrossRef\]](#)
24. Wu, Z.; Yu, S.; Li, T. A Meta-Model-Based Multi-Objective Evolutionary Approach to Robust Job Shop Scheduling. *Mathematics* **2019**, *7*, 529. [\[CrossRef\]](#)
25. Zahid, T.; Agha, M.H.; Schmidt, T. Investigation of surrogate measures of robustness for project scheduling problems. *Comput. Ind. Eng.* **2019**, *129*, 220–227. [\[CrossRef\]](#)
26. Wu, Z.; Sun, S.; Yu, S. Optimizing makespan and stability risks in job shop scheduling. *Comput. Oper. Res.* **2020**, *122*, 104963. [\[CrossRef\]](#)
27. Zheng, P.; Zhang, P.; Wang, J.; Zhang, J.; Yang, C.; Jin, Y. A data-driven robust optimization method for the assembly job-shop scheduling problem under uncertainty. *Int. J. Comput. Integr. Manuf.* **2020**, *33*, 1–16. [\[CrossRef\]](#)
28. Yang, Y.; Huang, M.; Wang, Z.Y.; Zhu, Q.B. Robust scheduling based on extreme learning machine for bi-objective flexible job-shop problems with machine breakdowns. *Expert Syst. Appl.* **2020**, *158*, 113545. [\[CrossRef\]](#)
29. Dui, H.; Xu, Z.; Chen, L.; Xing, L.; Liu, B. Data-Driven Maintenance Priority and Resilience Evaluation of Performance Loss in a Main Coolant System. *Mathematics* **2022**, *10*, 563. [\[CrossRef\]](#)
30. Mao, X.; Lou, X.; Yuan, C.; Zhou, J. Resilience-Based Restoration Model for Supply Chain Networks. *Mathematics* **2020**, *8*, 163. [\[CrossRef\]](#)

31. Qi, Q.; Meng, Y.; Zhao, X.; Liu, J. Resilience Assessment of an Urban Metro Complex Network: A Case Study of the Zhengzhou Metro. *Sustainability* **2022**, *14*, 11555. [\[CrossRef\]](#)
32. Xiao, S.; Wu, Z.; Sun, S.; Jin, M. Research on the Dual-resource Constrained Robust Job Shop Scheduling Problems. *J. Mech. Eng.* **2021**, *57*, 227–239.
33. Wang, K.; Choi, S.H.; Lu, H. A hybrid estimation of distribution algorithm for simulation-based scheduling in a stochastic permutation flowshop. *Comput. Ind. Eng.* **2015**, *90*, 186–196. [\[CrossRef\]](#)
34. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Kan, A.R. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discret. Math.* **1979**, *5*, 287–326.
35. Leon, V.J.; Wu, S.D.; Storer, R.H. Robustness measures and robust scheduling for job shops. *IIE Trans.* **1994**, *26*, 32–43. [\[CrossRef\]](#)
36. Hazır, Ö.; Haouari, M.; Erel, E. Robust scheduling and robustness measures for the discrete time/cost trade-off problem. *Eur. J. Oper. Res.* **2010**, *207*, 633–643. [\[CrossRef\]](#)
37. Goren, S.; Sabuncuoglu, I.; Koc, U. Optimization of schedule stability and efficiency under processing time variability and random machine breakdowns in a job shop environment. *Nav. Res. Logist.* **2012**, *59*, 26–38. [\[CrossRef\]](#)
38. Larranaga, P.; Lozano, J.A. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*; Springer: New York, NY, USA, 2002.
39. Xiao, S.; Sun, S.; Guo, H.; Jin, M.; Yang, H. Hybrid Estimation of Distribution Algorithm for Solving the Stochastic Job Shop Scheduling Problem. *J. Mech. Eng.* **2015**, *51*, 27–35. [\[CrossRef\]](#)
40. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. In *IEEE Transactions on Evolutionary Computation*; IEEE: Manhattan, NY, USA, 2002; Volume 6, p. 15.
41. Valledor, P.; Gomez, A.; Puente, J.; Fernandez, I. Solving Rescheduling Problems in Dynamic Permutation Flow Shop Environments with Multiple Objectives Using the Hybrid Dynamic Non-Dominated Sorting Genetic II Algorithm. *Mathematics* **2022**, *10*, 2395. [\[CrossRef\]](#)
42. Muth, J.F.; Thompson, G.L. *Industrial Scheduling*; Prentice-Hall: Hoboken, NJ, USA, 1963.
43. Lawrence, S. *Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques (Supplement)*; Carnegie-Mellon University: Pennsylvania, PA, USA, 1984.
44. Ahmadi, E.; Zandieh, M.; Farrokh, M.; Emami, S.M. A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. *Comput. Oper. Res.* **2016**, *73*, 56–66. [\[CrossRef\]](#)