

Article

Data-Mining-Based Real-Time Optimization of the Job Shop Scheduling Problem

Anran Zhao, Peng Liu ^{*}, Xiyu Gao, Guotai Huang, Xiuguang Yang, Yuan Ma, Zheyu Xie and Yunfeng Li

School of Mechanical and Aerospace Engineering, Jilin University, Changchun 130025, China

^{*} Correspondence: liupeng@jlu.edu.cn

Abstract: In the job-shop scheduling field, timely and proper updating of the original scheduling strategy is an effective way to avoid the negative impact of disturbances on manufacturing. In this paper, a pure reactive scheduling method for updating the scheduling strategy is proposed to deal with the disturbance of the uncertainty of the arrival of new jobs in the job shop. The implementation process is as follows: combine data mining, discrete event simulation, and dispatching rules (DRs), take makespan and machine utilization as scheduling criteria, divide the manufacturing system production period into multiple scheduling subperiods, and build a dynamic scheduling model that assigns DRs to subscheduling periods in real-time; the scheduling strategies are generated at the beginning of each scheduling subperiod. The experiments showed that the method proposed enables a reduction in the makespan of 2–17% and an improvement in the machine utilization of 2–21%. The constructed scheduling model can assign the optimal DR to each scheduling subperiod in real-time, which realizes the purpose of locally updating the scheduling strategy and enhancing the overall scheduling effect of the manufacturing system.

Keywords: pure reactive scheduling; subscheduling period; dispatching rule (DR); decision tree; scheduling model

MSC: 90-08

Citation: Zhao, A.; Liu, P.; Gao, X.; Huang, G.; Yang, X.; Ma, Y.; Xie, Z.; Li, Y. Data-Mining-Based Real-Time Optimization of the Job Shop Scheduling Problem. *Mathematics* **2022**, *10*, 4608. <https://doi.org/10.3390/math10234608>

Academic Editor: Shih-Wei Lin

Received: 3 October 2022

Accepted: 30 November 2022

Published: 5 December 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 is an important part of job shop manufacturing systems and a decisive factor affecting production efficiency [1]. As a classic combinatorial optimization problem, job shop scheduling has attracted scholars for more than 60 years because of its NP-hard characteristics [2]. Industries with high discreteness and strong customization, such as crystal ornaments and toys, often lead to unbalanced technological processes in the production process due to the variety of technological processes and large differences in processing time. Good scheduling strategies play a huge role. Due to disturbances encountered being more frequent and diverse in manufacturing systems, it is clear that static scheduling models can no longer fully meet manufacturing requirements. Dynamic scheduling needs to consider to machine degradation and the random arrival of products in the production process [3]. These disturbances are difficult to predict precisely or predictably in the early stage of scheduling, which may make the original scheduling strategy unsuitable. Due to the existence of the abovementioned disturbances, the complexity of scheduling increases rapidly. Therefore, research on the dynamic job shop scheduling problem (JSSP) remains a topic of considerable interest in the manufacturing industry, with strong application value and theoretical significance.

In increasingly complex manufacturing systems, uncertain events cause information asymmetry between the actual production process and pre-established scheduling strategies, affecting the execution of the pre-established scheduling strategies. Predictable disturbances can be dealt with in advance while unpredictable disturbances, such as the

random arrival of jobs, can have a catastrophic impact on production efficiency in the case of improper scheduling. Dealing with these unpredictable disturbances rapidly and accurately ensures the smooth execution of the manufacturing system production process and is the key to improving production efficiency. However, the following challenges are encountered in the current research process:

- (1) The metaheuristics [4], heuristic methods [5], and exact methods [6] used to optimize scheduling problems are often affected by computing resources, thus yielding poor performance of the scheduling strategies generated in a short time [7]. Moreover, these methods cannot evolve with state changes in the manufacturing system, and they are insufficient to cope with disturbance factors in the manufacturing system continuously and efficiently.
- (2) Makespan as the scheduling goal has an extensive engineering background and is widely used in academic research and industrial practice [8]. However, in actual production processes, the value-added time only accounts for a small part of the production cycle, and most of the wasted time is nonvalue-added time, such as waiting and storage [9]. Reducing the proportion of nonvalue-added time is an effective way to optimize makespan and improve production efficiency.
- (3) The unknown moment when the job is released to the manufacturing system and the unpredictability of its processing attributes make it impossible to obtain all of the job information that needs to be processed at the initial moment of the manufacturing system; thus, it is impossible to optimize the JSSP from a global perspective.

To address these challenges, the computational efficiency of the solution and the adaptive capacity of the obtained scheduling strategies should be considered. In this study, we apply data mining (DM) techniques to scheduling and mine scheduling knowledge from historical data to construct a scheduling model. This method can reduce the time consumption of the solution process and provide a basis for the intelligent generation of workshop scheduling strategies. The optimization problem of the entire manufacturing system can be divided into multiple subsystem optimization problems to be solved [10], thereby reducing the complexity of the entire manufacturing system. Scheduling newly released jobs with subsystem units can avoid or mitigate disturbances to the state of the entire manufacturing system. Moreover, the acquired scheduling knowledge can be used to partially update the manufacturing system scheduling strategies to achieve the goal of enhancing the overall scheduling efficiency of the manufacturing system.

Based on the above discussion, this study proposes a novel purely reactive scheduling method for optimizing the JSSP to overcome the negative impact of the randomness of job arrivals on production efficiency. The production scheduling cycle of the entire manufacturing system is divided into several scheduling subcycles, and scheduling strategies are generated at the beginning of each scheduling subcycle. The generated scheduling strategies are used to schedule newly released jobs and jobs transferred from previous processes in the previous scheduling subperiod. In executing the scheduling strategies in each scheduling subperiod, the original data containing scheduling knowledge are collected, and DM technology is used to mine the obtained scheduling knowledge. A scheduling model is then constructed in a data-driven manner, and a scheduling strategy is generated for each scheduling subcycle in real-time. The implementation of the above process can not only quickly generate a scheduling strategy at the beginning of each sub-scheduling cycle but can also actively schedule the inserted new jobs, avoiding the negative impact of the random release of a job on the manufacturing process.

The remainder of this paper is organized as follows. Section 2 summarizes research related to the JSSP and analyzes the optimization problem considered in this study. Section 3 analyzes the optimization problem considered in this study. Section 4 proposes the overall concept and method for solving the considered problem. Section 5 sets up an experiment to verify the proposed method. Finally, the results are summarized, and future prospects are discussed.

2. Review of the Literature

This section provides a review of numerous studies, summarizes gaps in previous research, and continues to optimize the scheduling problem on the basis of previous studies. The summary of previous studies on scheduling problems and scheduling methods is shown in Table 1 and is described in detail in the following three sections.

Table 1. Summary of the literature.

Literature	Scheduling	Description
[11–16]	Static scheduling	Propose advanced heuristic/metaheuristic algorithms to solve for better performance scheduling policies.
[17–23]	Robust scheduling	Construct a robust model in the early stages of scheduling to take into account possible perturbation problems.
[24–28]	Rescheduling	Monitor the moment a disturbance appears or the scheduling policy degrades and update the original scheduling policy with the set scheduling method.
[29–32]	Predictive reactive	Predict possible disturbances in real-time as production proceeds, and proactively update scheduling policies when disturbances are predicted.
[33–38]	Pure reactive scheduling	Generate scheduling policies in real-time and dynamically based on the current job information of the production system and the current production status without the need to generate scheduling policies in advance.

2.1. Data Mining and Its Application to the JSSP

Data mining (DM) is a technology that utilizes statistical computing, data management, machine learning, and artificial intelligence to extract hidden patterns or relationships from a large amount of data and apply them to future analysis [1]. It has been widely used in manufacturing, the service industry, and other fields [39,40]. With the ongoing application of DM technology in all walks of life, a large amount of experience has been accumulated. Scheduling is an integral part of manufacturing systems, and DM technology has been widely used to optimize production scheduling problems. DM technology provides new possibilities for solving dynamic scheduling problems and enhancing the responsiveness of manufacturing systems as it can rapidly obtain scheduling strategies. Alica et al. [39] proposed that DM technology could identify meaningful patterns and optimize problems encountered during the operation of a manufacturing system. Metan et al. [41] combined simulations, DM, and statistical process control chart technology to develop a real-time DR scheduling system to reduce the delivery delay time of jobs in a manufacturing system. Zahmani et al. extracted potential knowledge from scheduling strategies obtained using a genetic algorithm to reduce the computational load [42]. Shahzad et al. obtained a set of DR optimization job delays based on DM technology and confirmed their effectiveness in a static environment [43].

The above research shows that DM technology has been widely applied to solve scheduling problems in processing workshops and can directly generate scheduling strategies; however, the literature has mainly focused on static scheduling problems. In these cases, all of the job information to be scheduled is known, and the scheduling strategy is formulated before production starts. However, it is difficult to satisfy the production requirements in dynamic environments with frequent disturbances.

2.2. Job Shop Scheduling

Job shop scheduling includes both static and dynamic scheduling, and static scheduling can obtain a scheduling policy with superior performance in specific scenarios and stable production systems. Wei et al. [11] mixed a variable neighborhood equilibrium optimizer and a sticky bacterium algorithm to solve a JSSP, which accelerated the solution speed; Yu et al. proposed an improved particle swarm algorithm and compared the genetic algorithm and particle swarm. Yuraszeck et al. [12] proposed a heuristic algorithm to

avoid falling into the local optimum when solving scheduling policies, which is suitable for solving small-scale scheduling problems. Szabó et al. [13] proposed a heuristic algorithm for Clique Search in Graphs of Special Class, which can obtain better scheduling policies than genetic algorithms. Luan et al. [14,15] successively improved the whale algorithm under different scheduling objectives and then obtained faster performance scheduling policies than variable neighborhood search methods and some scheduling benchmarks. Sauvey et al. [16] constructed a makespan as the scheduling objective for the job shop scheduling problem with mixed blocking constraints and used a mathematical model to solve the scheduling policy using Mosel Xpress software. It was able to obtain a scheduling strategy with superior performance in specific scenarios and stable production systems, but it did not consider the possible uncertainties in the production process and had limitations in scheduling production systems with poor stability.

Production environments are becoming increasingly complex and dynamic because manufacturing systems are often disturbed by real-time events, such as external order insertion and task changes during the production process. Traditional static scheduling generally cannot meet the production requirements of such systems, and research on dynamic scheduling problems is thus becoming increasingly crucial. At present, according to the literature, the primary methods for modeling and solving dynamic scheduling problems include robust scheduling, rescheduling, predictive reactive scheduling, and purely reactive scheduling.

The robust scheduling method is an optimization method that integrates uncertain factors into the scheduling model. The key to a robust scheduling method is the prediction of the occurrence of future disturbance factors. If the prediction is biased, two unfavorable outcomes can occur in the manufacturing system: First, if the predicted disturbance factor does not appear, it will lead to an increased idle time of processing equipment and result in a large number of wasted resources [17]; second, incorrect prediction of the disturbance factors can cause the performance of the original scheduling strategy to deteriorate such that it may no longer be suitable for the current production environment [18]. Xiao et al. [19] constructed a robust scheduling model to solve the dynamic scheduling problem of the job shop with a random processing time. Fantahun et al. [20] proposed a simulated annealing algorithm to better schedule irreplaceable operators in the production workshop and improve production efficiency. Taking makespan as the scheduling goal, Khurshid et al. [21] successively proposed an evolutionary algorithm and a hybrid evolutionary strategy to optimize the scheduling problem of the permutation flow shop. To address the dynamic scheduling problem of randomly arriving tasks, Zhou et al. proposed an event-triggered dynamic task scheduling method in which uncertain events were considered in the scheduling model [22]. Wang et al. proposed two robust scheduling formulas based on scenario sets and used an improved tabu search algorithm for their solution; they were then applied to an uncertain JSSP with the completion time as the performance index [23].

Rescheduling is a method of updating the original scheduling strategy when a disturbance event is encountered during the operation of a manufacturing system. When the product requirements or working conditions change, the current production information is input into the determined mathematical model or algorithm, and the scheduling strategy is regenerated [24]. In the dynamic scheduling process, the timing of rescheduling, performance of the initial scheduling strategy, guarantee of an efficient rescheduling model, and performance of generating new scheduling strategies are the keys to the scheduling method. Rescheduling is generally performed when disturbances are encountered, so disturbances are predicted in advance or monitored in real-time. Zhang et al. [25] proposed an improved Kalman algorithm that can solve problems in a shorter time than the genetic algorithm and can be used in dynamic scheduling. Gao et al. established a scheduling model for the new task disturbance problem encountered in a flexible job shop manufacturing system, which was divided into two stages: scheduling and rescheduling [26]. Yin et al. [27] constructed a fine-grained system state description model for timely response to the flexible job shop scheduling problem with cost loss as the scheduling objective. Vakhania et al. [28] consid-

ered the job release date and delivery date, constructed a dynamic scheduling framework, and proposed a polynomial solver algorithm to update the scheduling strategy within a short timeframe.

The predictive reactive scheduling method first generates an initial scheduling strategy then predicts disturbance factors in the actual production process and dynamically updates the scheduling strategy [29], mainly for predictable disturbance events such as the machine status. Ji and Qiu used DM technology to explore the potential failure modes of workshop equipment, predict the probability of machine failure when processing current or future tasks, and provide references for scheduling and rescheduling [30,31]. Zhang et al. [32] integrated a convolutional neural network and improved the imperialist competition algorithm to overcome the negative impact of machine failure on the production process, which had a very good reference value for dealing with predictable failures. This method differs from the robust scheduling method in that it uses real-time data to predict disturbances. The predictability of disturbances is the premise of using this method. The accuracy of the prediction is guaranteed. However, this method has limitations for unpredictable disturbances, such as the release of new jobs.

The purely reactive scheduling method applies the production information and current production status of the manufacturing system to generate a dynamic real-time scheduling strategy. Because the scheduling strategy does not need to be created in advance, this method is also called “online scheduling” [33]. A short calculation time along with the adaptive and self-learning abilities of scheduling strategies are the basis for the efficient application of purely reactive scheduling methods [34]. Multiagent scheduling is a representative method of purely reactive scheduling. Researchers have proposed agent-based scheduling systems [35,36]. Each agent aims to maximize its interests, leading to conflicts with other agents. Mezgebe et al. proposed a negotiation-based control method to address disturbances caused by machine failures in a manufacturing system and achieved significant improvements in reducing the makespan [37]. Zhang et al. designed a two-tier distributed dynamic workshop scheduling system with a workshop scheduling agent and a multiservice unit scheduling agent. The negotiation mechanism used DR, and the service unit scheduling used a multiagent scheduling method based on humoral immunity [38]. Due to the static characteristics of the negotiation mechanism and the algorithms used by each agent, this method easily led to insufficient adaptability in a dynamic environment. However, the purely reactive scheduling method has good application value for solving unpredictable disturbance factors, such as the release of new jobs, because of its strong reactivity to the state of the manufacturing system.

2.3. Dispatching Rules

A DR is a heuristic rule. Due to its simplicity, interpretability, and low computational load, it has a shorter execution time than heuristic and metaheuristic methods [44] and a solid ability to respond to dynamic events. Its working principle is as follows: Whenever a machine is idle, a priority function is used to prioritize the unscheduled jobs waiting to be processed on the machine, and the job with the highest priority is processed first. The parameters of the priority function are usually an attribute value of the job and the current state of the manufacturing system. It uses the current information to determine which jobs need to be scheduled and how these jobs are scheduled. Therefore, DRs have been widely and successfully used to solve dynamic scheduling problems. In addition, this method neither requires modifying the original scheduling nor increases production instability.

Two methods have been developed to design a DR with excellent performance: human intervention design [45] and intelligent algorithm automatic design [46,47]. The first method requires repeated trial and error testing, and the designer must have extensive experience. The second design process is complex and unexplainable. Although many DRs have been created and applied, there is currently no DR that can solve a variety of problems or deal with a variety of disturbances with consistently excellent performance [7,45,48]. Researchers have extensively compared the performance of DRs and demonstrated that

a suitable DR response can be obtained under different scheduling problems or disturbance situations.

In the literature, the application of DRs to optimize the JSSP is divided into two categories. The first category involves selecting different DRs based on different scheduling criteria. Kaban, Veronique, and Marko analyzed different scheduling criteria. The results demonstrated the performance of the DRs and confirmed that no DR could provide the best performance under all scheduling criteria [7,45,49]. The other category involves selecting different DRs or DR combinations for job shop manufacturing systems in different situations. Tavakkoli et al. considered 10 different DRs and compared their performance in different manufacturing systems using makespan as the scheduling criterion. Then, the best DR to deal with the JSSP was selected [50]. Azadeh et al. proposed assigning DRs on a machine basis in a manufacturing system, determining the job processing sequence to be processed, and using the DR combination to optimize the makespan criterion [8]. Metan et al. used the delay time as the scheduling criterion to select the optimal DR combination from a predetermined DR library to optimize the JSSP [41]. However, the above methods solve static scheduling problems, i.e., all of the job information is obtained before scheduling. When the job information cannot be obtained in advance, the performance of the DR assigned to a manufacturing system or machine cannot be determined. In other words, there is a lack of research on how to apply DRs to optimize the JSSP when a job shop manufacturing system encounters a random process, such as when the job release moment or its attribute information cannot be obtained in advance.

As discussed above, DM has been widely used to generate scheduling strategies and predict disturbances. However, the generated scheduling strategies are mostly used to solve static scheduling problems. Integrating metaheuristics and heuristic algorithms to optimize dynamic scheduling problems is mostly based on the premise that disturbances are predictable. In order to overcome unpredictable disturbances, such as the random arrival of jobs, this paper proposes a pure reactive scheduling method that can generate scheduling strategies actively and in real-time, quickly combining the advantages of data model generation scheduling strategies and DRs.

3. Mathematical Model Construction

Before constructing the mathematical model, the involved notations were first defined, as shown in Table 2.

In a job shop, the order in which jobs are processed is random. The process flow of each job is diversified and known when it is released into the workshop. Due to the different jobs, the priorities of the multiple operations that comprise each job will be different [51]. In the manufacturing process, if a disturbance is considered, the feasible scheduling strategies will increase sharply, and it is very difficult to select the optimal or near-optimal scheduling strategy.

The job shop scheduling objective set in this paper is to minimize makespan and maximize machine utilization, and the objective function is defined as shown in Equations (1) and (2). Equation (2) represents the average of the actual processing operation time $\sum_{j=1}^n p_{ij}$ of the machine in the job shop production system as a percentage of the machine operation time C_i .

$$\text{Minimum } C_{\max} = \max\{C_i | i = 1, 2, \dots, m\} \quad (1)$$

$$\text{Maximum } M_u = \frac{1}{m} \sum_{i=1}^m \frac{\sum_{j=1}^n p_{ij}}{C_i} \quad (2)$$

In constructing the mathematical model, two core parameters, rp_{ijki} and f_k , are introduced, and the expression of the set rp_k composed of rp_{ijki} is shown in Equation (3); the solution of B is shown in Equation (4) with an initial value of 0. Among them, if there are

pending operations before the machine, each operation flowing out of the machine will immediately flow into another operation; if not, the machine carries out idling.

$$rp_k = \{rp_{1j_{k1}}, rp_{2j_{k2}}, \dots, rp_{mj_{km}}\} \quad (3)$$

$$f_k = \begin{cases} 0 & \text{if } k = 0 \\ \min\{rp_k | rp_{ij_{ki}} \neq 0\} & \text{if } k \neq 1 \end{cases} \quad (4)$$

Table 2. Indexes and parameters.

Notations	Descriptions
Indexes	
i	Index of machine, $i \in \{1, 2, \dots, m\}$
j	Index of jobs, $j \in \{1, 2, \dots, n\}$
k	Index of the machine in the production system when it completes the current operation, $k \in \{0, 1, \dots, K\}$
Parameters	
C_i	Time for machine i to complete all tasks, the running time of machine i
o_{ij}	Machine i processes the operation of job j
C_{ij}	Completion time of operation o_{ij}
$C_{\max}$	Maximum completion time for jobs in the production system (makespan)
p_{ij}	Processing time required for operation o_{ij}
r_{ij}	The moment the operation o_{ij_k} is released to the machine i
j_k	The job to which the k th operation of the machine output in the production system belongs
j_{ki}	Job j output by machine j_k
o_{ij_k}	Machine i processes the operation of job j_k
$rp_{ij_{ki}}$	The remaining processing time of o_{ij_k} , if the machine is idle, $rp_{ij_{ki}} = 0$
rp_k	$rp_{ij_{ki}}$ on all machines in the production system at the k th presence with outflow operations
f_{Ki}	The moment when machine i finally flows out the last job
Δk_{ij}	Total number of outflows experienced by machine i in the course of completing operation o_{ij} of job j
Decision variables	
x_{ij}	whether the machine is processing operation o_{ij} , if yes, $x_{ij} = 1$, if no, $x_{ij} = 0$.
f_k	the k – th moment when at least one machine in the production system has completed the current job
k_{ij}	The index of the outflow operation o_{ij} of machine i

Because machines take different amounts of time to process the job at hand, one machine may complete multiple jobs while another machine may not have completed one yet. Therefore, during the job flow, when jobs are flowing out of the machine, some machines may have some time remaining for the processing of the job at hand, and the remaining processing time experienced by the job during the machine processing will constitute a set rp_k which is obtained by Equation (3). When a machine completes a job, if there is a pending job next to that machine, the pending job immediately begins to be processed, at which point the time remaining for the machine to process the job is equal to the processing time for that job. The machine with the minimum remaining processing time among all machines in the production system gives priority to outflow jobs, and the inflow and outflow of jobs from the machine are expressed as a flow process, and the k th flow process is represented by Equation (4). After all the jobs flow out of the system, the task is completed and the time of the last job flowing out is the completion time, and the time of the last job flowing out of machine i is the time of machine i completing the task. Thus, Equations (1) and (2) are transformed into Equations (5) and (6).

In this paper, when solving the scheduling strategy, because the process of job flow is determined by the remaining time that the machine processes the current job, the rela-

relationship between the time that the machine processes the job and the job flow is shown in Equation (8). The time for each job to flow from the inflow to the outflow machine is equal to the time for the machine to process the job. Each operation o_{ij} is completed in a process in which the outflow of jobs has already occurred in other machines in the system, thus establishing the relationship between the time p_{ij} for each operation o_{ij} to be processed and the job flow process in the system, as shown in Equation (8).

The scheduling criterion makespan can be described as the time when the last operation of the last job in all tasks flows out of the production system, and C_i can be described as the time when the last operation flows out of machine i . According to the “outflow” characteristic, the objective function (2) can be transformed into the form shown in Equation (5), the objective function (3) can be transformed into the form shown in (6), and the whole mathematical model is shown in Equations (5)–(11).

$$\text{Minimum } C_{\max} = \sum_{k=0}^K (f_{k+1} - f_k) \quad (5)$$

$$\text{Maximum } M_u = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{j=1}^n p_{ij}}{f_{Ki}} \quad (6)$$

Subject to

$$r_{ij} + p_{ij} \leq r_{(i+1)j} \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (7)$$

$$\sum_{k=k_{ij}-\Delta k_{ij}}^{k_{ij}} f_k - f_{k-1} = p_{ij} \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (8)$$

$$rp_{ij_{ki}} \leq p_{ij} \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (9)$$

$$rp_K = \sum_{i=1}^m rp_{ij_{Ki}} = 0 \quad j = 1, 2, \dots, n \quad (10)$$

$$\sum_{i=1}^m x_{ij} = 1, x_{ij} \in \{0, 1\} \quad j = 1, 2, \dots, n \quad (11)$$

Constraint (7) indicates that operation o_{ij} of job j is processed by machine i before its next operation $o_{(i+1)j}$ is released to machine $i + 1$; Constraint (8) ensures that all operations of all jobs will be completed in the production system; Constraint (9) ensures that the remaining processing time of operation o_{ij} during the flow of the production system does not exceed the total processing time; Constraint (10) ensures that each operation o_{ij} of each job j can be processed, determines the index K value, and then solves for the scheduling target value; Constraint (11) indicates that only a single operation o_{ij} of job j can be processed by the machine in the same moment.

To better explain the above mathematical model, we take four machines as an example. Information on the jobs to be completed is shown in Table 3, and the PS column in the Table 3 indicates the process flow of the job. At the beginning of scheduling, machine 1 has job 1 and job 5 next to it, and according to the selected scheduling method, machine 1 processes job 1 first, $x_{11} = 1$, $x_{15} = 0$; machine 2 has job 2 next to it, $x_{22} = 1$; machine 3 has job 3 next to it, $x_{33} = 1$; and machine 4 has job 4 next to it, $x_{44} = 1$. No jobs are flowing from the machine in the production system, $f_k = 0$. At this point, the remaining processing time for machine 1 processing job 1 is 14, the remaining processing time for machine 2 processing job 2 is 50, the remaining processing time for machine 3 processing job 3 is 28, and the remaining processing time for machine 4 processing job 4 is 24. Therefore, $rp_0 = \{14, 50, 28, 24\}$, $\min rp_0 = 14$, and the production system starts to run. The first job to flow out of the machine is machine 1, the time of flow out is 14, $f_1 = 14$, job 1 flows out from machine 1 to the side of machine 2, and job 5 flows into machine 1. At this point, $rp_0 = \{77, 36, 14, 10\}$, $\min rp_0 = 10$, $k_{ij} = 1$, and $\Delta k_{ij} = 1$ and continues to iterate.

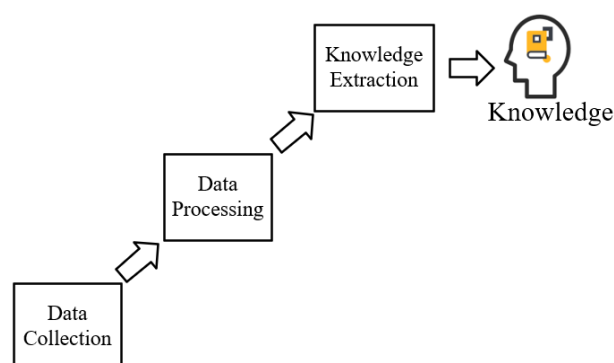
Table 3. Job shop production system with four machines and five operations.

Job	Machine 1	Machine 2	Machine 3	Machine 4	PS
1	14	87	31	43	1→2→4→3
2	90	50	60	56	2→1→3→4
3	21	46	28	55	3→4→2→1
4	81	54	44	24	4→2→1→3
5	77	65	32	76	1→3→4→2

In this study, the job shop system only considers two production factors: machines and materials. Only one machine is responsible for each process, and any technological process can be allowed. The moment when job i is released to the manufacturing system is random, and the processing information of the job before its release is unknown. The movement time of the job in the manufacturing system, machine degradation, and failures are not considered. When the job is released to the manufacturing system, its process flow and operating time on each machine become known. Due to the randomness of job release and its unpredictable characteristics, it is impossible to obtain a suitable scheduling strategy in the early stage of scheduling. The purpose of this paper is to reasonably solve the disturbance of job release randomness.

4. Proposed Approach

In this study, the implementation process of the proposed method is based on the general process of data mining technology. The original data with scheduling knowledge are initially obtained based on historical production data. Second, the available data with scheduling knowledge suitable for machine learning algorithms are obtained after data processing. Subsequently, the scheduling knowledge contained in the data is made explicit. Finally, a scheduling model is constructed, and the scheduling problem of the manufacturing system is optimized through the application of the model. The overall process of the proposed method is shown in Figure 1. The method is based on data collection and processing; knowledge extraction is the key to the proposed method; knowledge is obtained to solve the manufacturing system scheduling problem. To realize the above process, the following steps are performed: (1) relevant production data for the manufacturing system are collected, the scheduling goals are clarified, and factors affecting the pros and cons of the selected scheduling target values are analyzed to determine the optimal scheduling problem; (2) scheduling strategies to eliminate or reduce the influence of factors affecting the pros and cons of the scheduling target value (determined in step (1)) in the production scheduling process are determined to obtain the original data with scheduling knowledge; (3) the data foundation is consolidated for knowledge extraction to obtain the available data with scheduling knowledge; (4) a scheduling model that can generate scheduling strategies in real-time is constructed to allow for iterative scheduling of waiting jobs in the production process. The specific implementation process is described below.

**Figure 1.** Overall process of the proposed method.

4.1. Scheduling Subperiod Division

The purpose of dividing scheduling subperiods is to achieve local scheduling of known jobs and to generate scheduling strategies in the early stage of subscheduling. Due to the differences in different manufacturing systems and the uncertainty of the time of encountering disturbances, the division of scheduling subperiods should be active, dynamic, and carried out with the production process.

This study considers two goals in the production schedule. The first goal is to minimize the makespan of the manufacturing system and reduce the job waiting time, and the second goal is to continue increasing the actual load rate of each machine after obtaining the first target optimal value to reduce the idle time of the machine. The four-machine job shop problem is considered an example. As shown in Figure 2, the filled circles represent operation o_{ij} of the job waiting to be processed, and the open circles represent the next operation $o_{(i+1)j}$ of the job waiting to be processed. Assuming that Figure 2 represents a moment (m_t) in the job shop, it can be seen that there are pending operations in queues 0, 1, and 2. Machines 0, 1, and 2 will change to the working state at the next moment, queue 4 is empty, and machine 3 remains idle at the next moment. In the absence of new jobs released into the manufacturing system, if machine 0 processes operation o_{11} first, machine 3 will change from idle to working at $m_t + p_{11}$; if machine 0 processes operation o_{42} first, machine 3 will change from idle to working at $m_t + p_{42} + p_{11}$. The order in which machine 0 processes the operations in queue 0 will not only affect the load rate of machine 3 but also increase the makespan of the entire manufacturing system. Using specific methods or rules to determine the processing sequence for operations waiting to be processed in front of a machine can reduce the makespan of the manufacturing system and increase the actual load rate of the machine.

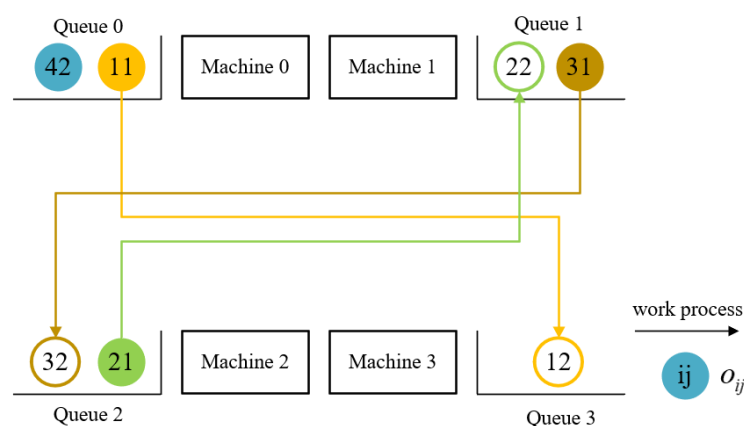


Figure 2. Job processing flow.

When the state of the machine changes from working to idle, the selected DR sorts the jobs waiting for processing next to each machine according to parameter values such as the job attribute or process flow, determines the priorities of the operations to be processed, and processes the operations with the highest priority until the queue is empty. Many studies have shown that no DR can effectively optimize multiple scheduling problems [52], and no DR can be adapted to different manufacturing systems under the same scheduling problem [7]. However, frequently switching DRs for scheduling problems to adapt to the scheduling of different manufacturing systems can effectively improve the production efficiency of the manufacturing system [53]. In the production process of the entire manufacturing system, taking the machine as a unit, the overall running time of each machine is iteratively divided into multiple scheduling subperiods, and all of the scheduling subperiods constitute the manufacturing system scheduling period. A DR generation scheduling strategy with excellent performance is selected at the beginning of

each scheduling subperiod. Therefore, determining the scheduling subperiod is the key to using DRs to optimize production scheduling.

The number of scheduling subperiods in the manufacturing system scheduling period is the number of times DRs are assigned. Each DR assignment is defined as a decision point. For the running process of a machine, the time between two adjacent decision points is a scheduling subperiod. As shown in Figure 3, a machine in the manufacturing system is considered an example. At m_1 , there are two operations waiting to be processed in front of the machine. At this moment, a DR with good performance is selected to determine the priorities of the two operations to be processed, the first decision is made, and m_1 is decision point 1. When the machine processes the operation, it is immediately transferred to the operation queue of jobs to be processed in front of the machine that processes the next operation. After both operations are processed, the time shifts to m_2 . Between m_1 and m_2 , as a result of operations completed by other machines and jobs newly released to the manufacturing system, a new queue of jobs to be processed will be formed next to the machine. At this time, the second decision is made, and m_2 is decision point 2. This decision point will schedule the newly released job in the manufacturing system, avoiding the adverse effect of the release of a new job disturbing the overall operation process of the manufacturing system. The time between m_1 and m_2 is defined as a scheduling subperiod of the manufacturing system scheduling period. The above process is iterated continuously, and each scheduling subperiod of the machine operation process can be obtained. The method for determining the scheduling subperiod for the unit machine in the production scheduling system is the same as that described above. After determining each scheduling subperiod, selecting a DR with good performance at the beginning of each scheduling subperiod is the most critical issue for optimizing production scheduling.

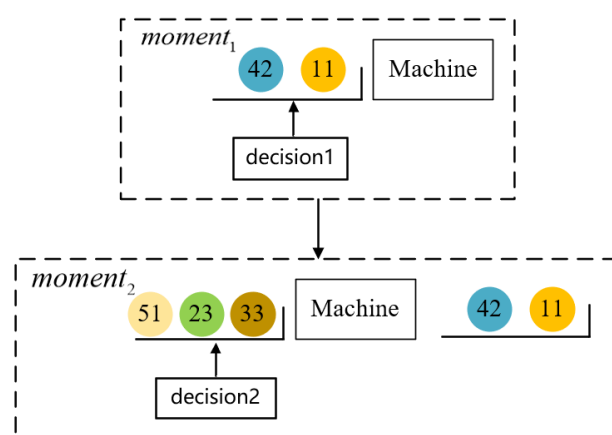


Figure 3. Determination of the scheduling subperiod.

4.2. Schedule Data Collection Method

Each scheduling subperiod has job processing data and scheduling strategies, and these data contain rich scheduling knowledge. Simulation technology is a powerful tool for evaluating the efficiency of different scheduling strategies. These methods can simulate processes in a short time without changing the state of physical entities to obtain the results of applying different scheduling strategies in the future [52]. As shown in Figure 3, the determination of the scheduling strategies for each scheduling subperiod affects the production efficiency of the entire manufacturing system.

To collect the original data with scheduling knowledge, simulation technology was used in this study to determine the optimal DR for each scheduling subperiod. In this process, it is necessary to simulate the production scheduling cycle of the entire manufacturing system. A DR is suitable for the scheduling subperiod only if it results in better performance in the entire production scheduling period. In obtaining the DR combination, the equation for solving the simulation number (N_s) of the manufacturing system production cycle is

shown in Equation (12), where N_{sp} represents the number of subperiods and N_{dr} represents the number of DRs that can be selected. As N_s and N_{sp} increase, the number of simulations will increase exponentially.

In the actual application process, when facing a complex manufacturing system, an excessive number of simulations will be required, which will lead to a time-consuming method of selecting the optimal DR combination using the simulation method; as a result, it will be difficult to ensure the fluency of the production process. Metan et al. [41] noted that a long time is required to calculate simulation results when real-time decision-making is necessary; thus, the simulation method will become useless. Accelerating the selection of the optimal DR combination to ensure a smooth production process is the key to using DRs to solve the dynamic scheduling problem in complex manufacturing systems. In addition, when applying simulation technology, all of the production information in the manufacturing system should be known. Therefore, it is not feasible to apply simulation technology to time consumption and generation-scheduling strategies. However, simulation technology can be used to obtain the original data containing scheduling knowledge from historical production data. The NP-hard characteristics of the JSSP result in relatively good DR combinations. The better the performance of the DR combination, the better the collected data will be and the higher the quality of the scheduling knowledge contained in the data.

$$N_s = (N_{dr})^{N_{sp}} \quad (12)$$

4.3. Construction of the Scheduling Model

To solve the scheduling problem of rapidly assigning a DR in the scheduling subcycle, based on the original data that contain scheduling knowledge obtained using simulation technology, a machine learning algorithm is used to mine the scheduling knowledge contained in the operation of the entire manufacturing system, and a DR assignment model is constructed. Thus, a solution to the above problems is achieved.

As shown in Figure 4, this study divides the process of constructing the assigned DR model into three modules: data collection, simulation, and learning. The data collection module collects job attribute data and processes the data in the job manufacturing system; the collected data is stored in database 1. The simulation module obtains the data containing scheduling knowledge. The realization process selects the optimal DR combination for each manufacturing system. The optimal DR combination is combined with the data collected by the data collection module to produce data with scheduling knowledge; these data are stored in database 2. The learning module accesses the knowledge in the learning data and builds a DR assignment model. The assigned DR model can be directly used in the scheduling process for a manufacturing system.

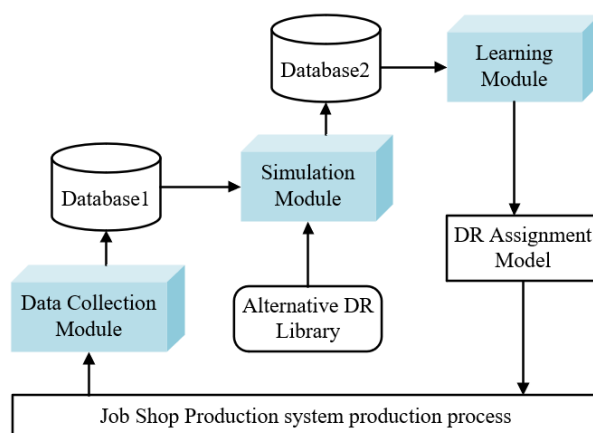


Figure 4. Assigned DR model construction process.

The scheduling model is the core of the method proposed in this study. This ensures that a scheduling strategy can be generated in real-time. The construction process for the scheduling model is shown in Figure 5, including the collection of raw data containing scheduling knowledge, the assembly of high-quality available data, and the training and prediction of multiclassification models. The specific implementation process is as follows.

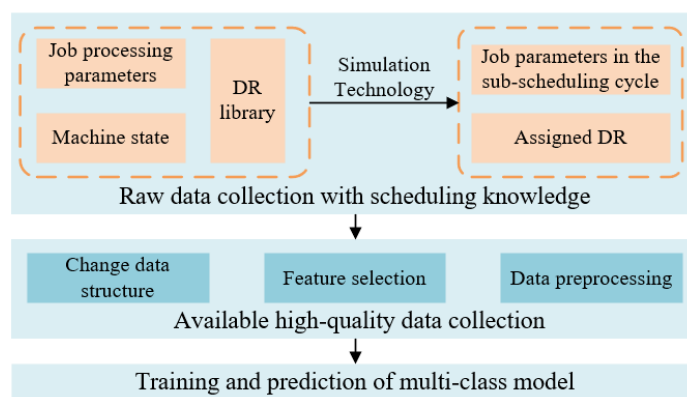


Figure 5. Decision model construction process.

Collect raw data with scheduling knowledge: Historical production data include processing parameters, such as the processing time of the job on each machine, the process flow in the manufacturing system, the state of each machine in the manufacturing system, and a DR library composed of multiple DRs. Of these, this study only considers the idle and working states of the machine. The above data can be collected directly; however, they contain no scheduling knowledge. As described in the previous section, the determined scheduling decision moment divides the scheduling period of the entire manufacturing system into multiple scheduling subperiods. The job processing data are linked to the DRs in the DR library based on simulation technology. The relevant parameter data of the job in each scheduling subperiod and the assigned DR in the scheduling subperiod are combined to form an instance sample, and the data contain rich scheduling knowledge. The job parameter data in the scheduling subcycle are defined as the attribute data of the instance samples, and the assigned DR is defined as the label of the instance samples. Simulating multiple manufacturing systems in the same category can ensure sufficient historical instance samples. Because the times when the job is released and the job release moment are unknown, the collection of original data with scheduling knowledge is based on historical production data.

Collect high-quality available data: Before using machine learning algorithms to mine the knowledge contained in the data, it is necessary to ensure the availability of data and their suitability for input into machine learning classification algorithms; it is also necessary to ensure the data are of a high quality so that the constructed model will have good performance. For the former, this study adopts the method of changing the data structure because the attribute data of the collected sample data are n rows and m columns. If n is equal to 1, the sample data of this instance do not contain scheduling knowledge because when a scheduling subperiod only includes one job to be scheduled, the effect of assigning any DR is indistinguishable. When n is greater than 1, the sample data contains scheduling knowledge, but this example sample is not suitable as the input data for the machine learning classification algorithm because the attribute data structure of the input data of the machine learning classification algorithm should be one row and m columns. Therefore, the structure of the attribute data of the collected data with scheduling knowledge needs to be changed, and there is a need to transform the original data structure $(n * m) \times d$ into an $n * (m + 1)$ structure. This study employs data feature selection and data preprocessing methods for this purpose. The data preprocessing includes removing duplicate values, filling in missing values, and deleting outliers. The feature selection process is mainly used

to eliminate features that are of no value in the data and prevent such features from affecting the performance of the model, such as the time when a job is released to the system.

Training and prediction of multiclassification models: A typical problem in supervised learning is classification [53]. In classification problems, supervised learning is used to construct a classifier that can label new instances based on experience with known labeled instance samples. The problem in this study is a classification problem because the label of the example sample comprises multiple values, and the constructed decision model is a multiclassification model [54]. The DT algorithm has been widely used in the establishment of classification models due to its ease of understanding, fast calculation speed, high accuracy, and lack of required domain knowledge or parameter assumptions [55]. Based on the above advantages, this study uses a DM algorithm to build a decision model to realize the real-time assignment of DRs to each scheduling subperiod of the manufacturing system scheduling period. The available high-quality data are divided into training and test sets in proportion to the construction process. The training data are used to build the model, and the test data are used to evaluate the performance of the trained model. The cooperation between the training and prediction processes ensures a stronger prediction performance of the model.

4.4. Application of the Scheduling Model

After the construction of the decision model is completed, the next step is the application of the decision model. According to the method described in the previous section, the model is used at each decision time. The processing parameters of the job to be scheduled on the machine at the decision time comprise the data input to the decision model before processing. If there is no job to be scheduled on the machine at the decision time, the machine state becomes idle, and the new decision time becomes the time before the job is released or transmitted to the machine. If there is only one job to be scheduled on the machine at the decision time, the decision model is not used, and the job is processed directly by the machine. If there are multiple jobs to be scheduled on the machine at the decision time, the parameter data of the jobs are combined to create the original data input to the decision model. After processing the data using the same method as when the model was constructed, the data are processed to produce usable, high-quality data that can be directly input into the decision model. The decision model then outputs the DR, which generates the scheduling strategy for the scheduling subperiod.

5. Experiments and Results

5.1. Experiment Setup

In this paper, the experiment used an 8G running memory, i5cpu personal computer. To verify the feasibility of the proposed method and the efficiency of the constructed decision-making model, a set of control experiments were designed and executed for different manufacturing systems. The method of integrating Taillard and Marko to generate the JSSP is consistent with actual-scale industrial problems [49,51]. Because the output scheduling strategy time of the decision model is significantly shorter than the processing time of the job operation, the precondition was to ignore the decision time of the decision model. Before generating the scheduling problem, the following assumptions were made:

- the delivery dates of all operations will be ignored;
- the transportation time between machines is zero;
- no faults occur during machine processing.

In the specific generation method, the number of jobs, attributes of the jobs, number of machines, and technological process of the jobs comprised a manufacturing system. With the goal of minimizing the makespan and maximizing the actual utilization of machines, the scheduling of each manufacturing system was defined as a JSSP. In the manufacturing system, the job attributes included the job processing time on the machine and the moment when the job is released to the manufacturing system. The processing time, pt_{ij} , of each operation on each machine required to complete the job satisfied the geometric distribution

of $(0, \phi)$, and the size of ϕ controlled the difference in the processing time of the operation to complete the job on the machine. The greater the value of ϕ , the greater the time difference between machine processing operations, and thus, the greater the heterogeneity of the jobs. The release moment, r_j , when the job is released to the manufacturing system was random, but the upper limit should be the total production time of the manufacturing system. This study assumes that the release moment, r_j , satisfied the geometric distribution of $(0, \phi)$. The value of ϕ controlled the load rate of the manufacturing system. The smaller the value of ϕ , the higher the load rate of the manufacturing system. The number of machines was n , and the machines were numbered as $\{0, 1, \dots, n-1\}$. The process flow of the job involved selecting one or more machines in the manufacturing system and rearranging and combining them; each machine only processed each job once. From the above, the generated manufacturing system included the following four categories: high job heterogeneity and a high manufacturing system load rate, low job heterogeneity and a low manufacturing system load rate, high job heterogeneity and a low manufacturing system load rate, and low job heterogeneity and a high manufacturing system load rate.

Table 4 gives an example of the manufacturing system data for four machines and 10 jobs. Each row represents a job. The values in column R_j represent the release moment, r_j , when the job is released to the manufacturing system; the empty cells in column R_j represent that the job to be processed on the machine was transferred to the previous machine after processing. The values in column PT_j represent the time for the job to be processed on the machine. If the value of PT_{ij} is 0, job j does not need to be processed on machine i . The value in column M_i represents the order of operations for processing the job. For example, for job 0, the job was released to the system at time 65, and the process flow proceeded through M_3 , M_2 , M_0 , and M_1 in sequence; the processing times on M_3 , M_2 , M_0 , and M_1 were 26, 5, 14, and 87, respectively.

Table 4. Manufacturing system data example (four machines and ten jobs).

No.	M0	R0	PT0	M1	R1	PT1	M2	R2	PT2	M3	R3	PT3
0	2		14	3		87	1		5	0	65	26
1	2		90	3		50	1		31	0	0	5
2	0	0	21	1		6	2		10	3		43
3	1		1	0	4	54	3		28	2		56
4	2		77	1		51	0	0	44	3		15
5	0	6	32	3		46	2		80	1		24
6	2		11	0	4	9	1		48	3		77
7	0	6	90	1		93	3		78	2		58
8	3		60	1		79	0	37	27	2		33
9	3		49	0	0	40	2		83	1		54

After the historical production data of the manufacturing system were collected, a DR library was constructed to schedule the jobs in the manufacturing system, and a group of highly adaptive DR combinations was selected from the DR library to guide the generation of scheduling strategies for the manufacturing system. The parameter value of the DR was obtained through the job properties. According to the characteristics of the generative manufacturing system in this study, the job attributes were as follows:

- Processing time (PT): the time required for the operation of the job to be processed on the machine;
- Processing sequence (PS): the total number of operations required by the job in the manufacturing system;
- Total processing time (TPT): the total time the job was processed in the manufacturing system; the total processing time, TPT_j , of job j is given in Equation (13):

$$TPT_j = \sum_{i=1}^m pt_{ij} \quad (13)$$

- Remaining total processing time (RPT): the total remaining processing time of the job in the manufacturing system; the remaining total processing time, RPT_j , of job j is shown in Equation (14), where $\hat{j}$ represents the number of operations to be performed in the current job:

$$RPT_j = \sum_{j=\hat{j}}^n pt_{ij} \quad (14)$$

- Remaining total processing sequence (RPS): the number of operations remaining in the manufacturing system for the job; the remaining total processing sequence of job j is shown in Equation (15):

$$RPS_j = n - \hat{j} \quad (15)$$

Based on the attributes of the above job, 10 DRs were defined to create the single-parameter DR library, as summarized in Table 5. The first six DRs in Table 5 were static single-parameter DRs, and their parameter values did not change with the job process; the last four DRs were dynamic single-parameter DRs, and their parameter values did change with the job process. The constructed DR library contained static single-parameter DRs and dynamic single-parameter DRs, which ensures that the DRs can solve various dynamic scheduling problems.

Table 5. Single-parameter DRs.

No.	DR	Parameter	Description
1	SPT	Processing time	Priority processing for jobs with the shortest processing time
2	LPT	Processing time	Priority processing for jobs with the longest processing time
3	SPS	Processing sequence	Priority processing for jobs with the shortest processing sequence
4	LPS	Processing sequence	Priority processing for jobs with the longest processing sequence
5	STPT	Total processing time	Priority processing for jobs with the shortest total processing time
6	LTPT	Total processing time	Priority processing for jobs with the longest total processing time
7	SRPT	Total remaining processing time	Priority processing for jobs with the shortest total remaining processing time
8	LRPT	Total remaining processing time	Priority processing for the job with the longest total remaining processing time
9	SRPS	Total remaining processing sequence	Priority processing for jobs with the shortest total remaining processing sequence
10	LRPS	Total remaining processing sequence	Priority processing for jobs with the longest total remaining processing sequence

$$P = P_1 \times P_2 \quad (16)$$

Based on the current knowledge, no single-parameter DR can adapt to the scheduling process of an entire manufacturing system. The method for constructing a new DR proposed by Kaban [45] was applied to construct the DR library required in this study. The second category of DRs in the constructed DR library was composed of mixed-parameter, in which two job attributes were combined. The equation for solving the parameter value of the mixed-parameter DR is given in Equation (16), where P_1 is one value of a job attribute, and P_2 is another value of a job attribute. For example, P_1 is the value of the processing time PT, and P_2 is the value of the total processing time TPT; the value of $PT * TPT$ is the parameter value of the hybrid DR. If the value of $PT * TPT$ for the job waiting to be processed by the machine is small, the mixed-parameter DR corresponding to the priority being processed is S_PT_TPT . If the value of $PT * TPT$ for the job waiting to be processed by the machine is large, the mixed-parameter DR corresponding to the priority being processed is L_PT_TPT .

The job attributes and the hybrid DRs' parameters which calculated by Equation(16) are listed in Table 6 resulting in a total of 10 hybrid parameters. Different DRs were constructed by sorting the priority of the jobs according to the parameter values from small to large and from large to small, and each parameter value could create two mixed-parameter DRs.

Table 6. Mixed parameter DR.

Parameter	PT	PS	TPT	RPT	RPS
PT	×	✓	✓	✓	✓
PS	×	×	✓	✓	✓
TPT	×	×	×	✓	✓
RPT	×	×	×	×	✓
RPS	×	×	×	×	×

In summary, the DR library constructed in this study contains 10 single-parameter DRs and 20 mixed-parameter DRs, totaling 30 DRs.

$$\hat{P} = \max \left(\sum_j p_{0j}, \sum_j p_{1j}, \sum_j p_{2j}, \sum_j p_{3j} \right) \quad (17)$$

$$\varphi = \frac{\hat{P}}{\lambda} \quad (18)$$

To verify the performance of the proposed method for the four manufacturing system categories, 20 sets of data were generated for each manufacturing system category. Each manufacturing system contained four machines and 50 operations. The attribute data of each category of the manufacturing system were determined based on the geometric distribution in a specific interval, as summarized in Table 7. The equations for the upper limit value, φ , of the geometric distribution interval subject to the release moment of the job are given in Equations (17) and (18), where $\hat{P}$ represents the ideal makespan of the manufacturing system, which is equivalent to the maximum actual working time for a machine to complete all tasks in the manufacturing system.

Table 7. Attribute value ranges of a job in the manufacturing system.

No.	Category	R	PT
1	High job heterogeneity and high manufacturing system load rate	$[0, \hat{P}/2]$	$[1, 100]$
2	Low job heterogeneity and low manufacturing system load rate	$[0, \hat{P}]$	$[1, 100]$
3	High job heterogeneity and low manufacturing system load rate	$[0, \hat{P}/2]$	$[1, 1000]$
4	Low job heterogeneity and high manufacturing system load rate	$[0, \hat{P}]$	$[1, 1000]$

Because of the large number of different DR combinations in the entire manufacturing system, it was unrealistic to use simulation technology to search all of the possible DR combinations to obtain the optimal DR combination. Therefore, this study used a local search method to identify the optimal DR combination. To verify the performance of the searched DR combination, the minimum limit value of the makespan was considered. The value of the minimum makespan limit, $L_Makespan$, was equal to the value of $\hat{P}$.

The implementation tool of all the above experimental processes was Spyder software, and the programming language was Python. Among them, the DR in the DR library was written as a function that reproduced the production process of the job shop and simulated the performance of each DR combination, and automatically output the scheduling target value and collect data with scheduling knowledge.

5.2. Acquisition of Data with Scheduling Knowledge

The data with scheduling knowledge were obtained through simulation, and a DR was assigned to each subscheduling period by means of a full search, and then a DR combination

was assigned to the entire scheduling period. In this study, 10,000 DR combinations were randomly obtained from the DR library for each manufacturing system. As the scheduling subperiod was divided, the number of DRs in different DR combinations might have differed due to the influence of each DR on the scheduling subperiod. For 20 manufacturing systems generated under the same category, the best and worst DR combinations from 10,000 DR combinations were selected; the large differences between the optimal and worst scheduling objective values verify the sensitivity of the DR combinations to the optimal scheduling problem. Figures 6–9 show comparisons of the optimal makespan limit value, the makespan value under the optimal DR combination, and the makespan value under the worst DR combination as well as the sum of the best actual machine utilization rate under the optimal makespan value and the sum of the optimal actual machine utilization rate under the optimal makespan value for different categories of manufacturing systems. The makespan value of the optimal DR combination was close to or equal to the optimal limit value of the makespan, which confirms that the optimal DR combination among the 10,000 DR scheduling combinations exhibited excellent scheduling performance. However, it was also possible to continue expanding the DR search range to obtain the optimal DR in order to obtain a better DR combination. Figure 6 shows a comparison of the performance of different DR combinations for a high job heterogeneity and high manufacturing system load rate. The optimization range of the makespan value was 3–15%, and the improvement in the machine utilization of the manufacturing system was 3–17%. Figure 7 shows a comparison of the performance of different DR combinations for a low job heterogeneity and low manufacturing system load rate. The optimization range of the makespan value was 5–15%, and the improvement in the machine utilization of the manufacturing system was 2–16%. Figure 8 shows a comparison of the performance of different DR combinations for a high job heterogeneity and high manufacturing system load rate. The optimization range of the makespan value was 5–16%, and the improvement in the machine utilization of the manufacturing system was 2–21%. Figure 9 shows a comparison of the performance of different DR combinations for a high job heterogeneity and low manufacturing system load rate. The optimization range of the makespan value was 6–13%, and the improvement in the machine utilization of the manufacturing system was 3–14%. The above results verify the effectiveness of the scheduling method proposed in this study for handling uncertain disturbances in the job release.

Additionally, based on Figures 10–13 we analyzed the difference between using dispatching rule combination performance and genetic algorithm performance. The genetic algorithm used real number coding, and the individual coding adopted the method of random ordering of all processes. The mutation was a random exchange of the serial numbers of the two processes. The fitness was determined according to the calculation time of each individual coding. The evaluation indicators used mean deviation and coefficient of variation. The purpose of using the average deviation as the evaluation index was to express the difference in the overall performance of the two methods, and the purpose of using the coefficient of variation as the evaluation index was to express the overall stability of the performance of the two methods. When using makespan as the scheduling target, the average performance deviations of the combination of variable coefficient rules and the genetic algorithm under the four categories were −5.5, −16.8, −40.65, and −38.5, respectively. Continue to take the sum of machine utilization as the scheduling target. For the four categories of job shop scheduling systems, the average performance deviations of using the dispatching rule combination and the genetic algorithm were 0.006, 0.017, 0.028, and 0.021, respectively, which proves the overall superiority of using the dispatching rule combination to optimize the job shop scheduling problem. It can be seen from Table 8 that the overall performance stability of the dispatching rule combination optimization method was worse than that of the genetic algorithm when the makespan was used as the scheduling target. The combination of Figures 10–13 does not hide the performance of the dispatching rule combination optimization method, even if the genetic algorithm used in optimizing some job shop scheduling problems showed better performance; when we

continue to optimize machine utilization, the dispatching rule combination optimization method showed better stability.

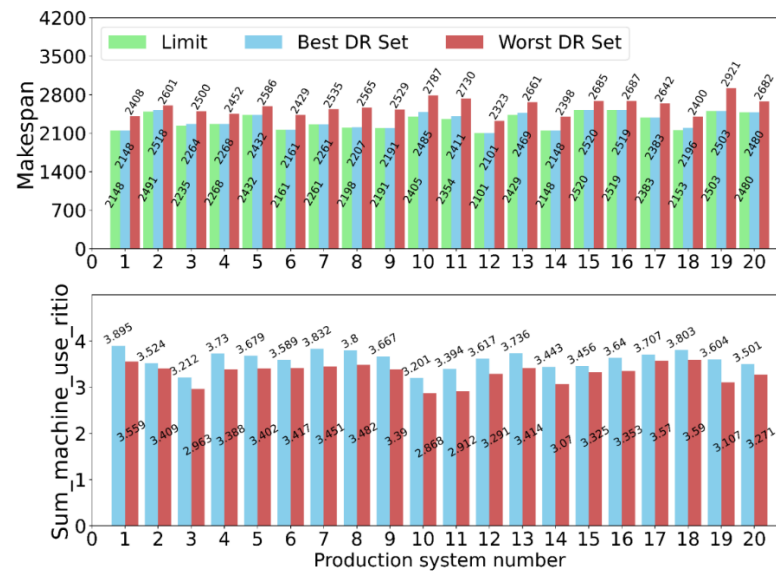


Figure 6. Low job heterogeneity and a high manufacturing system load rate.

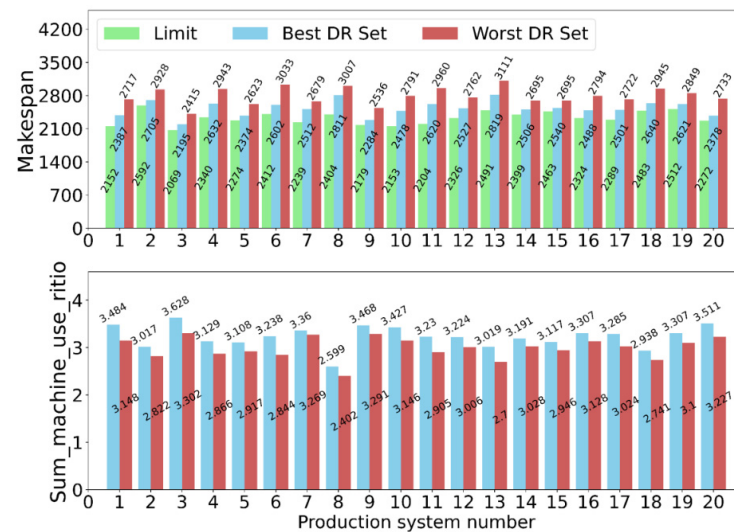


Figure 7. Low job heterogeneity and a low manufacturing system load rate.

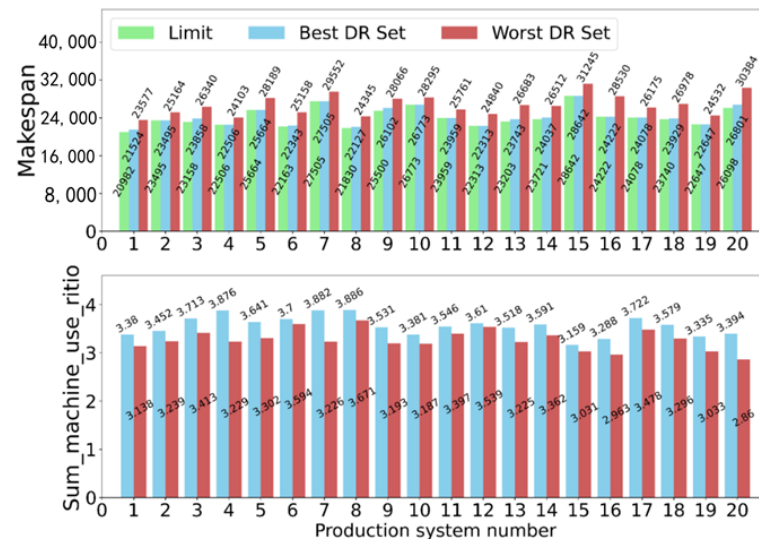


Figure 8. High job heterogeneity and a high manufacturing system load rate.

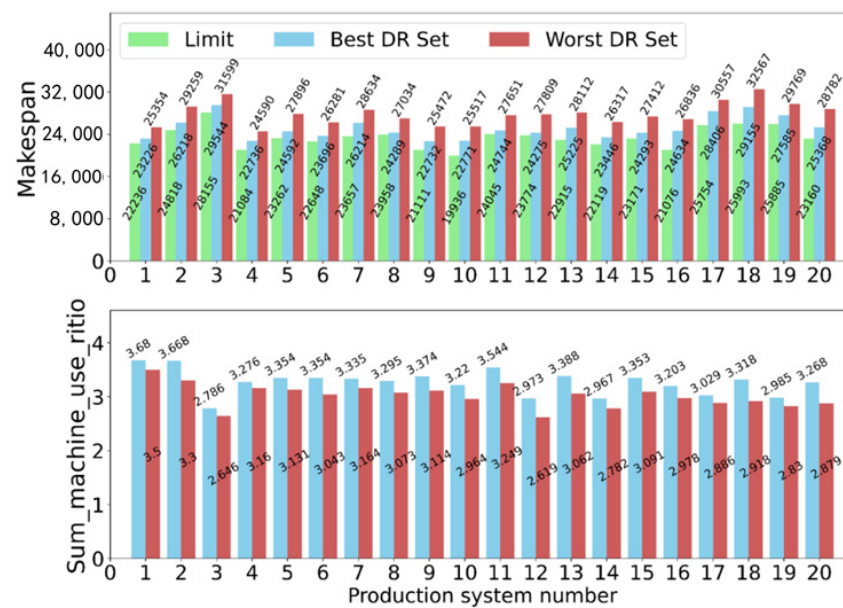


Figure 9. High job heterogeneity and a low manufacturing system load rate.

Table 8. Overall stability comparison of scheduling rule combination and genetic algorithm performance.

Scheduling Criterion	Method	Low Job Heterogeneity and High Load Rate	Low Job Heterogeneity and Low Load Rate	High Job Heterogeneity and High Load Rate	High Job Heterogeneity and Low Load Rate
Makespan	Best DR set	9.324	9.563	152.774	165.438
	GA	8.994	12.169	148.959	163.826
Machine use ratio	Best DR set	0.01	0.21	0.1	0.016
	GA	0.011	0.27	0.11	0.017

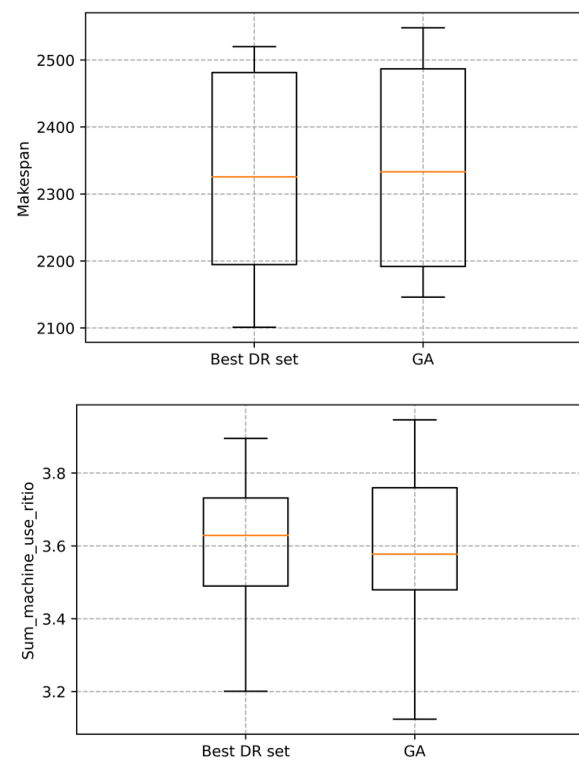


Figure 10. Low job heterogeneity and a high manufacturing system load rate.

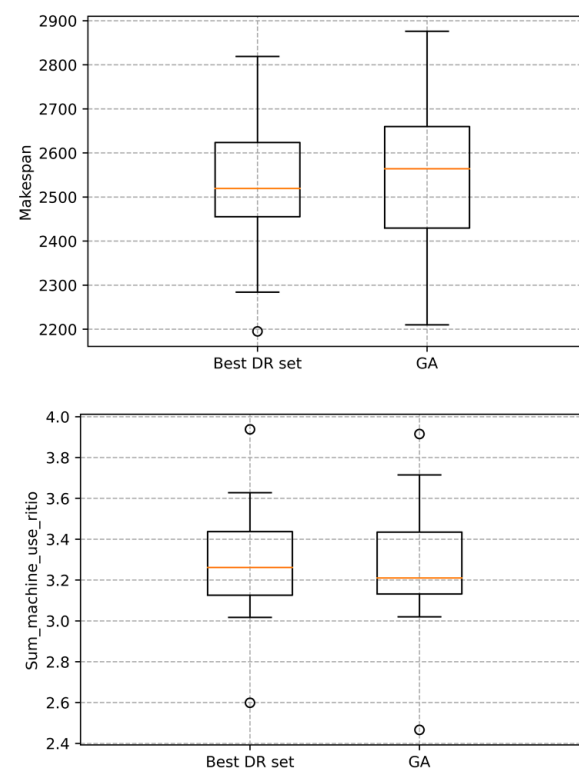


Figure 11. Low job heterogeneity and a low manufacturing system load rate.

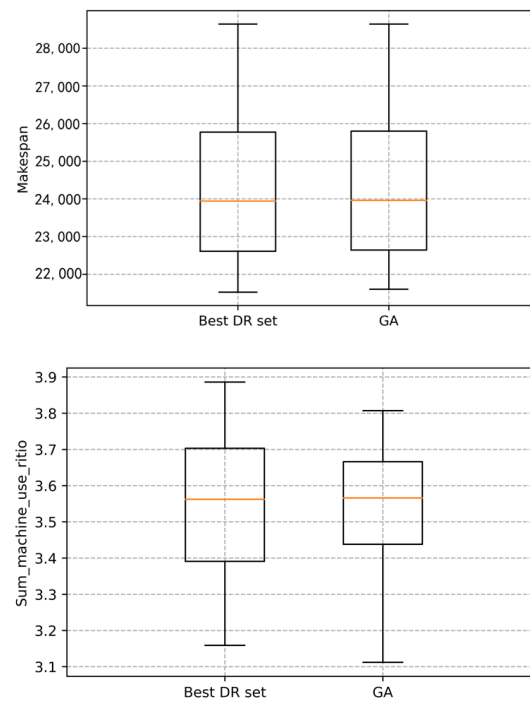


Figure 12. High job heterogeneity and a high manufacturing system load rate.

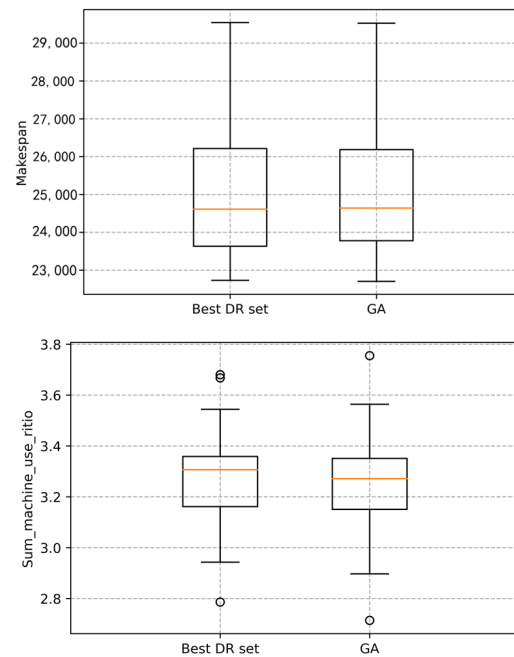


Figure 13. High job heterogeneity and a low manufacturing system load rate.

To sum up, the quality of the scheduling data with scheduling knowledge collected by the method in this paper can be guaranteed, thereby ensuring the availability and superiority of the constructed dispatching rule assignment model.

5.3. Data Preprocessing

After collecting the production data related to the manufacturing system and using simulation technology to search for the optimal DR combination for the manufacturing system from the DR library, the original data with scheduling knowledge could be col-

lected. The attributes of these data comprised the processing parameters of the jobs in the manufacturing system and the parameters of each DR.

The manufacturing system described in Table 4 is considered as an example. Three DRs (SPT, STPT, and STWR) were set in the DR library. After the simulation, they were divided into 29 scheduling subperiods, as listed in Table 9. Each sample data point represented a scheduling subperiod. The data structure contained sample data with a single attribute corresponding to one label and sample data with multiple attributes corresponding to one label. For the former, only one job was scheduled in the scheduling subperiod, which did not contain scheduling knowledge. The latter indicates that multiple jobs were scheduled in the scheduling subperiod, which contain scheduling knowledge. Because the data input to the machine learning algorithm should comprise a label corresponding to a row of attribute data, this type of datum cannot be directly input into the machine learning algorithm. Before using the DM algorithm to build a decision model, it is necessary to process the original data with scheduling knowledge using the methods of high-quality data collection described in the previous section. Given the characteristics and structure of the original data, the specific processing flow is described below.

Table 9. Raw data with scheduling knowledge (four machines and 10 jobs).

No.	M0	R0	PT0	M1	R1	PT1	M2	R2	PT2	M3	...	DR	No.
0	0	0	21	1		6	2		10	3	...	STWR	0
1	3		49	0	0	40	2		83	1	...	SPT	1
2	2		77	1		51	0	0	44	3	...	STPT	2
3	2		90	3		50	1		31	0	...	STWR	3
4											...	STPT	4
5	0	6	32	3		46	2		80	1	...	SPT	5
	0	6	90	1		93	3		78	2	...		
6											...	STWR	6
7	0	0	21	1		6	2		10	3	...	SPT	7
	2		11	0	4	9	1		48	3	...		
	1		1	0	4	54	3		28	2	...		
8	3		49	0	0	40	2		83	1	...	SPT	8
9	2		90	3		50	1		31	0	...	STPT	9
	3		60	1		79	0	37	27	2	...		
10	2		14	3		87	1		5	0	...	STPT	10
	0	6	32	3		46	2		80	1	...		
11	0	0	21	1		6	2		10	3	...	SPT	11
	2		11	0	4	9	1		48	3	...		
	3		49	0	0	40	2		83	1	...		
....													
29	0	6	90	1		93	3		78	2	...	SPT	29

The data feature selection process is illustrated in Figure 14. Because assigning DRs and then generating the scheduling strategy is based on the machine as the unit and dividing the scheduling subperiod is also based on the machine, the decision-making times on each machine were independent of each other. The attribute data of the original data with scheduling knowledge were grouped by machine and processed in the same manner. For the original data with scheduling knowledge in this study, the release time attribute column (columns R0, R1, R2, and R3) and machine columns (columns M0, M1, M2, and M3) were removed, and the grouped data were combined vertically. In this process, the number (quantity) of jobs in each scheduling subperiod was added manually, and the resulting data were defined as the data after feature selection.

The second step was a transformation of the data structure. Before the transformation, data preprocessing operations were performed to delete abnormal sample data that did not contain scheduling knowledge. Then, the data with scheduling knowledge were transformed into a structural form in which a row of attribute data corresponded to the label data. Commonly used mathematical and statistical methods include the sample mean,

sample standard deviation, coefficient of variation, and classic machine learning dimensionality reduction algorithms. In this study, the PCA algorithm was used, transforming the original data structure $(n * m) \times d$ into a $n * (m + 1)$ structure. Finally, the available data with scheduling knowledge were preprocessed, including the elimination of repeated values and deletion of outliers. The outlier deletion process employed the 3σ principle.

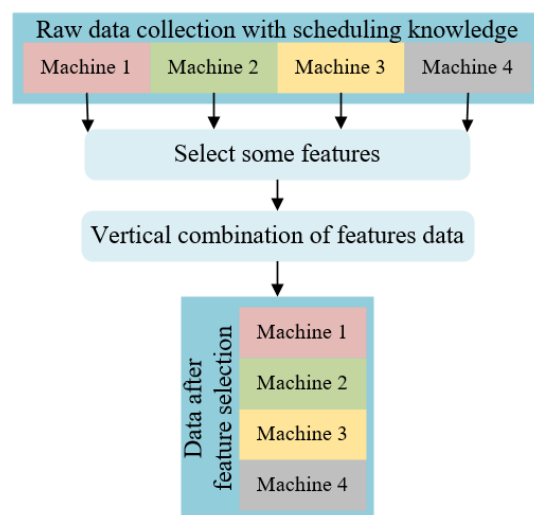


Figure 14. Feature selection process.

5.4. The Construction of the Scheduling Model

After obtaining the available high-quality data, the data were divided into a training set and a test set at a ratio of 7:3, and the DM algorithm was used to build a multiclassification model. In constructing the model, a grid search method was used to obtain the optimal parameters of the DR algorithm. The accuracies of the decision-making models constructed for manufacturing systems of different categories obtained after five rounds of crossvalidation are listed in Table 10.

Table 10. Classification model accuracy results.

Production System Category	Low Job Heterogeneity, High Load Rate	Low Job Heterogeneity, Low Load Rate	High Job Heterogeneity, High Load Rate	High Job Heterogeneity, Low Load Rate
Acc. (%)	79.96	77.99	85.92	83.81

5.5. Analysis of Model Performance

To verify the effectiveness of the constructed decision model for solving the scheduling problem in response to the disturbance factor of the job release moment with unknown released job information, 20 sets of data were collected for each manufacturing system category and randomly combined into 10,000 DR combinations. Simulation technology was used to search for the optimal DR combination with the goal of maximizing the sum of the utilization rate of each machine in the makespan and the manufacturing system. Then, the constructed decision model was used to iteratively assign DRs at each decision moment in the production process, and the generated scheduling strategy was used to assign the jobs in the scheduling subperiod. After all of the processing tasks of the manufacturing system were completed, the makespan value and sum of the utilization rate of each machine in the manufacturing system were calculated.

Figures 15–18 show comparisons between the makespan values and total machine utilization values of the manufacturing system obtained under the optimal DR combination identified through global search and the DR combination produced by the iterative assignment of the decision-making model for different categories of manufacturing systems.

Although the effect is less than that of the optimal DR combination obtained through the global search, the difference is insignificant, and the effect is the same in some manufacturing systems. Thus, the proposed method can solve the disturbance problem of the new job release in the manufacturing system efficiently and in real-time. A comparison of the decision-making effect of the decision model and the optimal DR combination obtained through the global search is shown in Figure 19 for each type of manufacturing system (Figure 19a represents high job heterogeneity and a high manufacturing system load rate, Figure 19b shows low job heterogeneity and a low manufacturing system load rate, Figure 19c shows high job heterogeneity and a high manufacturing system load rate, and Figure 19d represents high job heterogeneity and a low manufacturing system load rate); the statistical analysis of the comparison is summarized in Table 11. For the manufacturing system category of low job heterogeneity and a high manufacturing system load rate, compared with the most optimal DR combination, over half of the manufacturing system makespan increase rates were less than 1%; the average increase rate was 1.059%, and the median was 0.298%. The attenuation rates of the sum of the utilization values for each machine in half of the manufacturing systems were less than 1%; the average attenuation rate was 1.218%, and the median was 1.041%. For the manufacturing system category of low job heterogeneity and a low manufacturing system load rate, the average increase rate of the makespan value compared to the optimal DR combination was 3.139%, and the median was 3.276%; the average attenuation rate of the sum of the utilization values of each machine was 2.993% with a median of 3.276%. For the manufacturing system category of high job heterogeneity and a high manufacturing system load rate, compared with the most optimal DR combination, nearly half of the manufacturing system makespan increase rates were less than 1%; the average increase rate was 2.287%, and the median was 1.979%. The attenuation rates of the utilization value for each machine in over half of the manufacturing systems were less than 1%; the average attenuation rate was 2.287%, and the median was 1.979%. For the manufacturing system category of high job heterogeneity and a low manufacturing system load rate, compared with the most optimal DR combination makespan value, the average increase rate was 2.836%, and the median was 1.772%. Compared with the most optimal DR combination, the average attenuation rate of the utilization value for each machine was 2.712%, and the median was 1.908%.

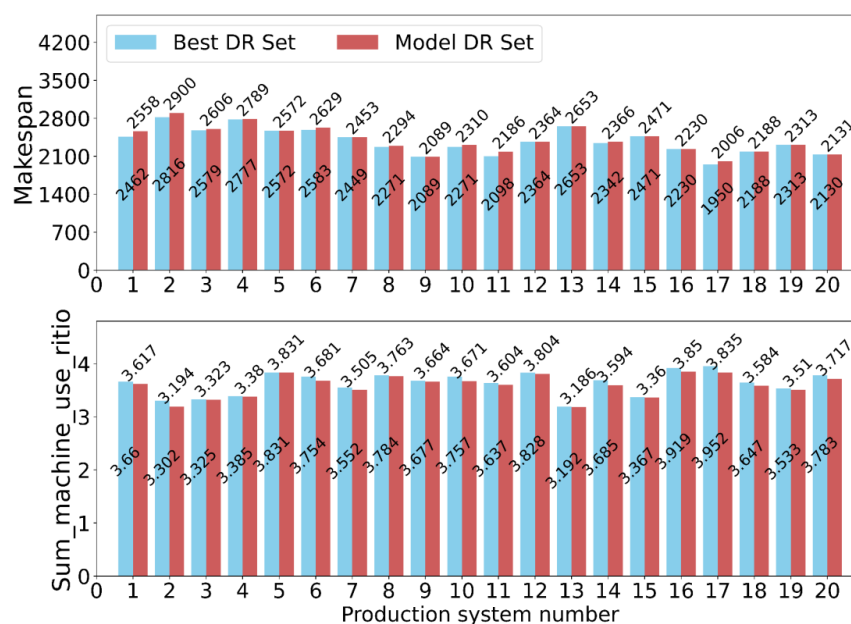


Figure 15. Low job heterogeneity and a high manufacturing system load rate.

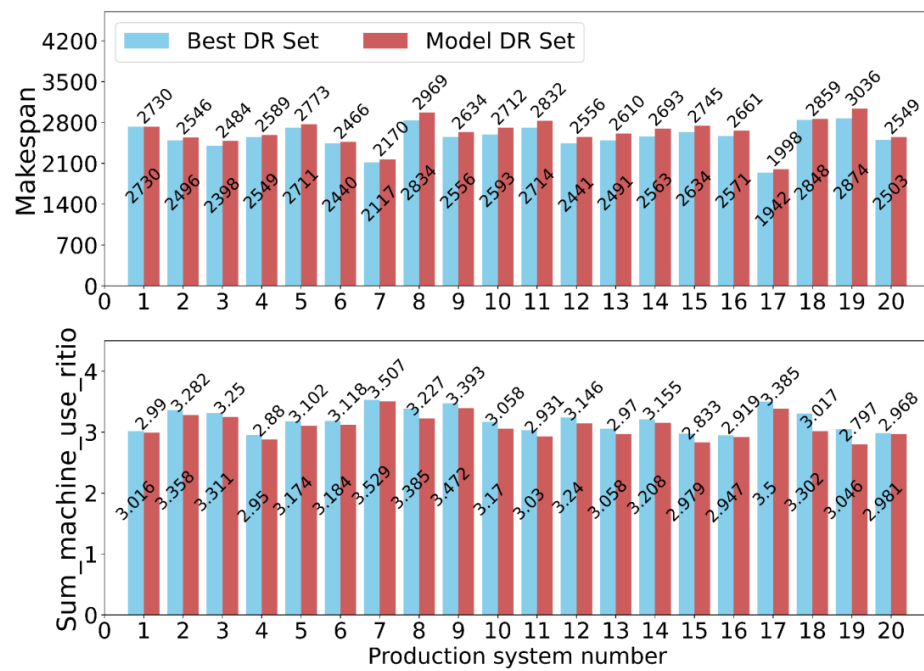


Figure 16. Low job heterogeneity and a low manufacturing system load rate.

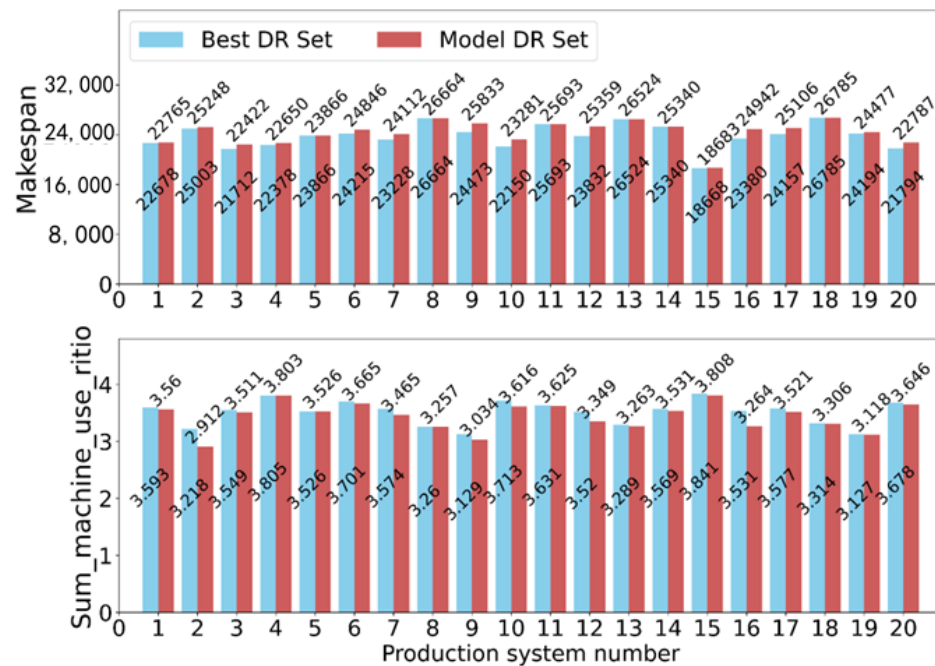


Figure 17. High job heterogeneity and a high manufacturing system load rate.

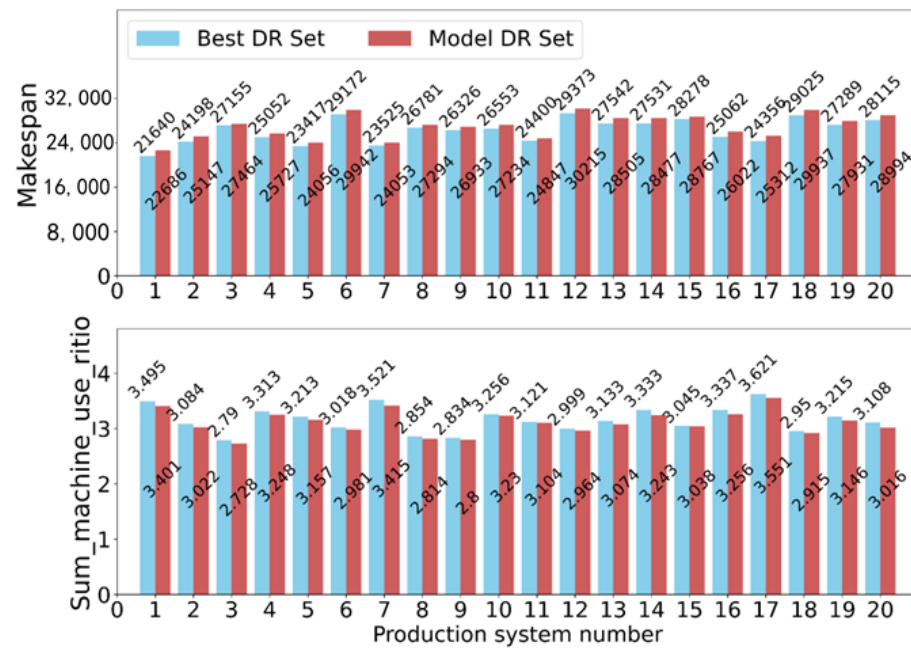


Figure 18. High job heterogeneity and a low manufacturing system load rate.

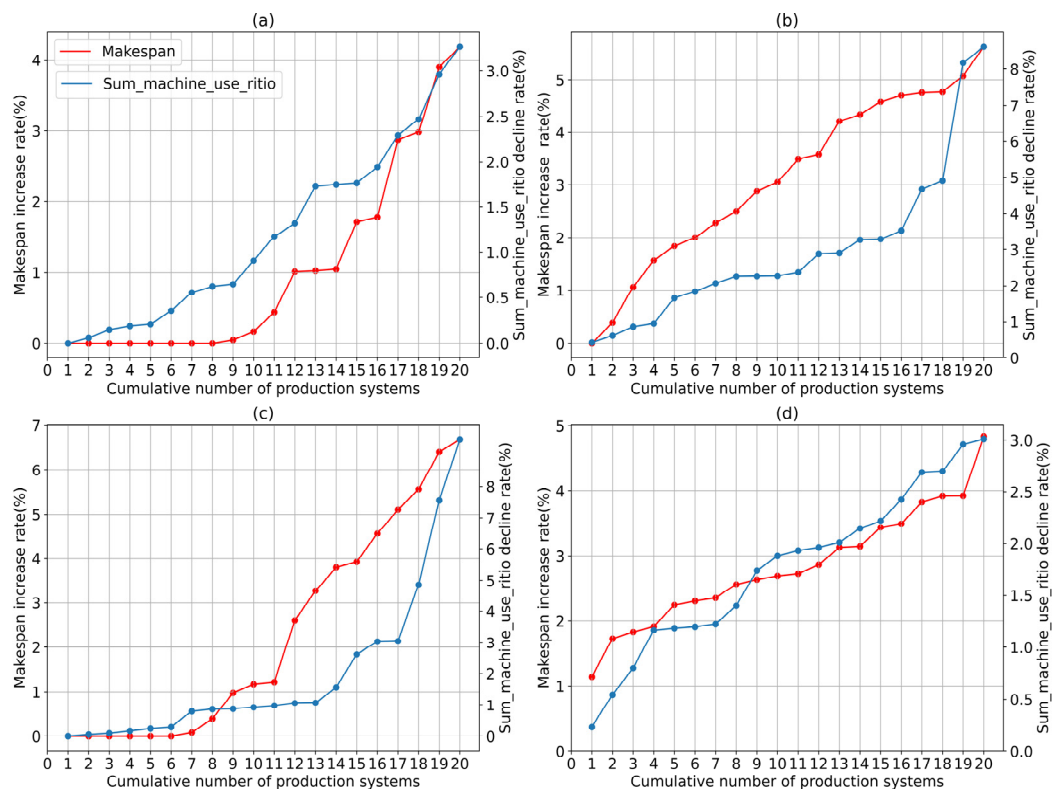


Figure 19. Comparison of the decision-making model and optimal DR combination results ((a) represents high job heterogeneity and a high manufacturing system load rate, (b) shows low job heterogeneity and a low manufacturing system load rate, (c) shows high job heterogeneity and a high manufacturing system load rate, and (d) represents high job heterogeneity and a low manufacturing system load rate).

Table 11. Statistics of the comparison between the decision model and optimal DR combination results.

Scheduling Criterion	Statistical Indicators	Low Job Heterogeneity and High Load Rate	Low Job Heterogeneity and Low Load Rate	High Job Heterogeneity and High Load Rate	High Job Heterogeneity and Low Load Rate
Makespan	Avg. (%)	1.059	3.139	2.287	2.836
	Med. (%)	0.298	3.276	0.193	2.712
Machine use ratio	Avg. (%)	1.218	2.993	1.979	1.772
	Med. (%)	1.041	2.324	0.946	1.908

In summary, the decision-making model constructed in this study had significant effects, and the decision-making effects of models for different categories were similar to those obtained using the optimal DR simulation. The time range for each decision and generation of a locally optimal scheduling policy was 0.12–0.21 s, which is much less than the processing time of the job on the machine. This verifies that the proposed method can provide an efficient and real-time response to the impact of the randomness of the new job release on the manufacturing system.

6. Conclusions

Data mining (DM) technology provides a novel and effective solution for real-time job shop disturbance problems. In this paper, a purely reactive scheduling approach is proposed based on the data mining process. Firstly, a novel DR combination method was used to collect data with scheduling knowledge; secondly, the collected data were used to construct a decision model that can assign DR in real-time; finally, a four-machine scale JSSP was used to verify the effectiveness of the proposed method. The main contributions of this paper are as follows.

- (1) A novel DR combinatorial optimization JSSP mixed-integer linear programming model was constructed, and a DR combinatorial optimization method is proposed.
- (2) A decision tree algorithm was used to train a decision model for assigning DRs from the collected data. The model assigned DRs at the beginning of each subscheduling cycle, which achieved an optimal overall scheduling policy by generating a locally optimal scheduling policy and overcame the adverse effects of the dynamic perturbation problem of midjob release.
- (3) The performance of the decision model DR combination method for assigning DR is similar under the scheduling objectives of makespan and machine utilization, demonstrating the superior performance of the scheduling policy generated using the DM technique.

However, the method proposed in this paper still has significant limitations. The construction of scheduling decision models using DM depends on the quality of the collected scheduling data, and further research is needed on how to generate scheduling data using the DR combination method; in terms of application scenarios, this paper only verifies the value of DM for optimizing dynamic scheduling problems on a classical job shop, and future research should focus on applications in more complex and wider production environments.

Author Contributions: Conceptualization, A.Z.; data curation, A.Z.; formal analysis, A.Z.; funding acquisition, P.L.; investigation, P.L.; methodology, P.L.; project administration, X.G.; software, X.G.; resources, G.H.; supervision, G.H.; validation, X.Y.; visualization, X.Y.; writing—original draft, Y.M.; writing—review and editing, Z.X.; writing—review and editing, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Jilin Scientific and Technological Development Program (Grant no. 20210201037GX) and Jilin Major Science and Technology Program (Grant no. 20210301037GX).

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

Conflicts of Interest: We would like to declare that no conflict of interest exists in the submission of this manuscript, and the manuscript was approved by all authors for publication. We also would like to declare on behalf of my coauthors that the work described is original research that has not been published previously and is not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

References

1. Liu, Z.F.; Chen, W.; Zhang, C.X.; Yang, C.B.; Cheng, Q. Intelligent scheduling of a feature-process-machine tool supernetwork based on digital twin workshop. *J. Manuf. Syst.* **2020**, *58 Part B*, 157–167. [\[CrossRef\]](#)
2. Garey, M.R.; Johnson, D.S.; Sethi, R. The Complexity of Flowshop and Job shop Scheduling. *Math. Oper. Res.* **1976**, *1*, 117–129. [\[CrossRef\]](#)
3. Barenji, A.V.; Barenji, R.V.; Roudi, D.; Hashemipour, M. A dynamic multi-agent-based scheduling approach for SMEs. *Int. J. Adv. Manuf. Technol.* **2017**, *89*, 3123–3137. [\[CrossRef\]](#)
4. Mohsen, Z. A heuristic algorithm for solving flexible job shop scheduling problem. *Int. J. Adv. Manuf. Technol.* **2014**, *71*, 519–528. [\[CrossRef\]](#)
5. Perez, F.; Raupp, F. A Newton-based heuristic algorithm for multi-objective flexible job-shop scheduling problem. *J. Intell. Manuf.* **2016**, *27*, 409–416. [\[CrossRef\]](#)
6. Veronique, S.; Nele, G.; Mario, V. A comparison of priority rules for the job shop scheduling problem under different flow time- and tardiness-related objective functions. *Int. J. Prod. Res.* **2012**, *50*, 4255–4270. [\[CrossRef\]](#)
7. Zhang, J.; Deng, T.M.; Jiang, H.F.; Chen, H.J.; Qin, S.F.; Ding, G.F. Bi-level dynamic scheduling architecture based on service unit digital twin agents. *J. Manuf. Syst.* **2021**, *60*, 59–79. [\[CrossRef\]](#)
8. Azadeh, A.; Negahban, A.; Moghaddam, M. A hybrid computer simulation-artificial neural network algorithm for optimisation of dispatching rule selection in stochastic job shop scheduling problems. *Int. J. Prod. Res.* **2012**, *50*, 551–566. [\[CrossRef\]](#)
9. Gutowski, T.; Murphy, C.F.; Allen, D.; Bauer, D. Environmentally benign manufacturing: Observations from Japan, Europe and the United States. *J. Clean. Prod.* **2005**, *13*, 1–17. [\[CrossRef\]](#)
10. Wang, H.Y.; Wang, J.C.; Xu, H.T.; Zhao, S.W. Distributed stochastic model predictive control for systems with stochastic multiplicative uncertainty and chance constraints. *ISA Trans.* **2021**, *121*, 11–20. [\[CrossRef\]](#)
11. Wei, Y.F.; Othman, Z.; Daud, M.K.; Yin, S.H.; Luo QFZhou, Y.Q. Equilibrium Optimizer and Slime Mould Algorithm with Variable Neighborhood Search for Job Shop Scheduling Problem. *Mathematics* **2022**, *10*, 4063. [\[CrossRef\]](#)
12. Yuraszcek, F.; Mejía, G.; Pereira, J.; Vilà, M. A Novel Constraint Programming Decomposition Approach for the Total Flow Time Fixed Group Shop Scheduling Problem. *Mathematics* **2022**, *10*, 329. [\[CrossRef\]](#)
13. Szabó, S.; Záválnij, B. Clique Search in Graphs of Special Class and Job Shop Scheduling. *Mathematics* **2022**, *10*, 697. [\[CrossRef\]](#)
14. Luan, F.; Cai, Z.Y.; Wu, S.Q.; Jiang, T.H.; Li, F.K.; Yang, J. Improved Whale Algorithm for Solving the Flexible Job Shop Scheduling Problem. *Mathematics* **2019**, *7*, 384. [\[CrossRef\]](#)
15. Luan, F.; Cai, Z.Y.; Wu, S.Q.; Qiang, S.; He, Y. Optimizing the Low-Carbon Flexible Job Shop Scheduling Problem with Discrete Whale Optimization Algorithm. *Mathematics* **2019**, *7*, 688. [\[CrossRef\]](#)
16. Sauvey, C.; Trabelsi, W.; Sauer, N. Mathematical Model and Evaluation Function for Conflict-Free Warranted Makespan Minimization of Mixed Blocking Constraint Job-Shop Problems. *Mathematics* **2020**, *8*, 121. [\[CrossRef\]](#)
17. Wang, B.; Xie, H.X.; Xia, X.D.; Zhang, X.X. A NSGA-II Algorithm Hybridizing Local Simulated-Annealing Operators for a Bi-Criteria Robust Job-Shop Scheduling Problem Under Scenarios. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 1075–1083. [\[CrossRef\]](#)
18. Shen, X.N.; Han, Y.; Fu, J.Z. Robustness measures and robust scheduling for multi-objective stochastic flexible job shop scheduling problems. *Soft Comput.* **2017**, *26*, 6531–6554. [\[CrossRef\]](#)
19. Xiao, S.C.; Wu, Z.G.; Dui, H.Y. Resilience-Based Surrogate Robustness Measure and Optimization Method for Robust Job-Shop Scheduling. *Mathematics* **2022**, *10*, 4048. [\[CrossRef\]](#)
20. Fantahun, M.D.; Dolapo, O.; Alebachew, D.Y. Mathematical model and simulated annealing algorithm for setup operator constrained flexible job shop scheduling problem. *Comput. Ind. Eng.* **2022**, *171*, 108487. [\[CrossRef\]](#)
21. Khurshid, B.; Maqsood, S.; Omair, M.; Sarkar, B.; Saad, M.; Asad, U. Fast Evolutionary Algorithm for Flow Shop Scheduling Problems. *IEEE Access* **2021**, *9*, 44825–44839. [\[CrossRef\]](#)
22. Zhou, L.F.; Zhang, L.; Sarker, B.R.; Laili, Y.J.; Ren, L. An event-triggered dynamic scheduling method for randomly arriving tasks in cloud manufacturing. *Int. J. Comput. Integr. Manuf.* **2018**, *31*, 318–333. [\[CrossRef\]](#)
23. Wang, B.; Wang, X.Z.; Xie, H.X. Bad-scenario-set robust scheduling for a job shop to hedge against processing time uncertainty. *Int. J. Prod. Res.* **2019**, *57*, 3168–3185. [\[CrossRef\]](#)
24. Wang, Y.R.; Wu, Z.L. Model construction of planning and scheduling system based on digital twin. *Int. J. Adv. Manuf. Technol.* **2020**, *109*, 2189–2203. [\[CrossRef\]](#)
25. Zhang, H.K.; Buchmeister, B.; Li, X.Y.; Ojstersek, R. Advanced Metaheuristic Method for Decision-Making in a Dynamic Job Shop Scheduling Environment. *Mathematics* **2021**, *9*, 909. [\[CrossRef\]](#)

26. Gao, K.Z.; Suganthan, P.N.; Chua, T.J.; Chong, C.S.; Cai, T.X.; Pan, Q. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion. *Expert Syst. Appl.* **2015**, *42*, 7652–7663. [\[CrossRef\]](#)
27. Yin, Y.; Kong, X.; Xia, C.Q.; Xu, C.; Jin, X. Low-Cost Emergent Dynamic Scheduling for Flexible Job Shops. *Mathematics* **2022**, *10*, 1873. [\[CrossRef\]](#)
28. Vakhania, N. Dynamic Restructuring Framework for Scheduling with Release Times and Due-Dates. *Mathematics* **2019**, *7*, 1104. [\[CrossRef\]](#)
29. Ouelhadj, D.; Petrovic, S. A survey of dynamic scheduling in manufacturing systems. *J. Sched.* **2009**, *12*, 417–431. [\[CrossRef\]](#)
30. Ji, W.; Wang, L.H. Big data analytics based fault prediction for shop floor scheduling. *J. Manuf. Syst.* **2017**, *43 Part 1*, 187–194. [\[CrossRef\]](#)
31. Qiu, Y.T.; Sawhney, R.; Zhang, C.Y.; Chen, S.; Zhang, T.; Lisar, V.G.; Jiang, K.; Ji, W. Data mining-based disturbances prediction for job shop scheduling. *Adv. Mech. Eng.* **2019**, *11*, 1. [\[CrossRef\]](#)
32. Zhang, G.H.; Lu, X.X.; Liu, X.; Zhang, L.T.; Wei, S.W.; Zhang, W.Q. An effective two-stage algorithm based on convolutional neural network for the bi-objective flexible job shop scheduling problem with machine breakdown. *Expert Syst. Appl. Int. J.* **2022**, *203*, 117460. [\[CrossRef\]](#)
33. Gupta, D.; Maravelias, C.T.; Wassick, J.M. From rescheduling to online scheduling. *Chem. Eng. Res. Des.* **2016**, *116*, 83–97. [\[CrossRef\]](#)
34. Dogan, A.; Birant, D. Machine learning and data mining in manufacturing. *Expert Syst. Appl.* **2021**, *166*, 114060. [\[CrossRef\]](#)
35. Zhou, Q.P.; Zhang, M.Z.; Ki-Hyung, B. Edge computing and financial service industry financing risk innovation based on data mining technology. *Pers. Ubiquitous Comput.* **2021**, *25*, 19. [\[CrossRef\]](#)
36. Gokhan, M.; Ihsan, S.; Henri, P. Real time selection of scheduling rules and knowledge extraction via dynamically controlled data mining. *Int. J. Prod. Res.* **2010**, *48*, 6909–6938. [\[CrossRef\]](#)
37. Zahmani, M.H.; Atmani, B. A Data Mining Based Dispatching Rules Selection System for the Job Shop Scheduling Problem. *J. Adv. Manuf. Syst.* **2019**, *18*, 35–36. [\[CrossRef\]](#)
38. Shahzad, A.; Mebarki, N. Data mining based job dispatching using hybrid simulation-optimization approach for shop scheduling problem. *Eng. Appl. Artif. Intell.* **2012**, *25*, 1173–1181. [\[CrossRef\]](#)
39. Sahin, C.; Demirtas, M.; Erol, R.; Baykasoğlu, A.; Kaplanoğlu, V. A multi-agent based approach to dynamic scheduling with flexible processing capabilities. *J. Intell. Manuf.* **2017**, *28*, 1827–1845. [\[CrossRef\]](#)
40. Xiong, W.; Fu, D.M. A new immune multi-agent system for the flexible job shop scheduling problem. *J. Intell. Manuf.* **2018**, *29*, 857–873. [\[CrossRef\]](#)
41. Zhou, Y.; Yang, J.J.; Zheng, L.Y. Multi-Agent Based Hyper-heuristics for Multi-objective Flexible Job Shop Scheduling: A Case Study in an Aero-engine Blade Manufacturing Plant. *IEEE Access* **2019**, *7*, 21147–21176. [\[CrossRef\]](#)
42. Mezgebe, T.T.; Bril El Haouzi, H.; Demesure, G.; Pannequin, R.; Thomas, A. Multi-agent systems negotiation to deal with dynamic scheduling in disturbed industrial context. *J. Intell. Manuf.* **2019**, *31*, 1–13. [\[CrossRef\]](#)
43. Dalila, F.; Mahdi, H.; Jos'e, F.G. A Hybrid Particle Swarm Optimization and Simulated Annealing Algorithm for the Job Shop Scheduling Problem with Transport Resources. *Eur. J. Oper. Res.* **2022**, *in press*. [\[CrossRef\]](#)
44. Durasevic M.Jakobovic, D. Comparison of solution representations for scheduling in the unrelated machines environment. In Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 30 May–3 June 2016; pp. 1336–1342. [\[CrossRef\]](#)
45. Rohmah, D.S.; Othman, Z.; Kaban, A.K. Comparison of dispatching rules in job-shop scheduling problem using simulation: A case study. *Int. J. Simul. Model.* **2012**, *11*, 129–140. [\[CrossRef\]](#)
46. Đurasević, M.; Jakobović, D. Creating dispatching rules by simple ensemble combination. *J. Heuristics* **2019**, *25*, 959–1013. [\[CrossRef\]](#)
47. Li, X.N.; Olafsson, S. Learning effective new single machine dispatching rules from optimal scheduling data. *Int. J. Prod. Econ.* **2010**, *128*, 118–126. [\[CrossRef\]](#)
48. Ozturk, G.; Bahadir, O.; Teymourifar, A. Extracting priority rules for dynamic multi-objective flexible job shop scheduling problems using gene expression programming. *Int. J. Prod. Res.* **2019**, *57*, 3121–3137. [\[CrossRef\]](#)
49. Đurasević, M.; Jakobović, D. A survey of dispatching rules for the dynamic unrelated machines environment. *Expert Syst. Appl.* **2018**, *113*, 555–569. [\[CrossRef\]](#)
50. Tavakkoli, M.R.; Daneshmand, M.M. A computer simulation model for job shop scheduling problems minimizing makespan. *Comput. Ind. Eng.* **2005**, *48*, 811–823. [\[CrossRef\]](#)
51. Taillard, E. Benchmarks for basic scheduling problems. *Eur. J. Oper. Res.* **1993**, *64*, 278–285. [\[CrossRef\]](#)
52. Boris, J.; Patnaik, G.; Lee, M.Y.; Young, T.; Leith, B.; Harms, F.; Schatzmann, M. Validation of an LES Urban Aerodynamics Model for Homeland Security. In Proceedings of the 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Expo-Sition, Orlando, FL, USA, 5–8 January 2009. [\[CrossRef\]](#)
53. Rokach, L. Ensemble-based classifiers. *Artif. Intell. Rev.* **2010**, *33*, 1–39. [\[CrossRef\]](#)
54. Read, J.; Reutemann, P.; Pfahringer, B.; Holmes, G. Meka: A Multi-label/Multi-target Extension to Weka. *J. Mach. Learn. Res.* **2016**, *17*, 667–671.
55. Kotsiantis, S.B. Decision trees: A recent overview. *Artif. Intell. Rev.* **2013**, *39*, 261–283. [\[CrossRef\]](#)