



HAL
open science

An Algorithm for Part Input Sequencing of Flexible Manufacturing Systems with Machine Disruption

Yumin He, Alexandre Dolgui, Milton Smith

► **To cite this version:**

Yumin He, Alexandre Dolgui, Milton Smith. An Algorithm for Part Input Sequencing of Flexible Manufacturing Systems with Machine Disruption. *Algorithms*, 2024, 17 (10), pp.470. 10.3390/a17100470 . hal-04925493

HAL Id: hal-04925493

<https://hal.science/hal-04925493v1>

Submitted on 21 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Article

An Algorithm for Part Input Sequencing of Flexible Manufacturing Systems with Machine Disruption

Yumin He ^{1,*}, Alexandre Dolgui ²  and Milton Smith ³¹ School of Economics and Management, Beihang University, Beijing 100191, China² IMT Atlantique, LS2N-CNRS, 44307 Nantes, France³ Department of Industrial Engineering, Texas Technology University, Lubbock, TX 79409, USA

* Correspondence: heyumin@buaa.edu.cn; Tel.: +86-010-18514218907

Abstract: Because disruption happens unpredictably and generates serious impact in supply chain and production environments in the real world, it is important to develop approaches to handle disruption. This paper investigates disruption handling in part input sequencing of flexible manufacturing systems (FMSs). An algorithm is proposed for FMS part input sequencing to handle machine breakage. Evaluation is performed for the proposed algorithm by simulation experiments and result analyses. Finally, conclusions are summarized with managerial implications discussed and further research works suggested.

Keywords: disruption handling; machine breakage; part input sequencing; FMS scheduling; dynamic scheduling; reactive scheduling

1. Introduction

Disruption events occur unpredictably in supply chain (SC) and production environments. Unexpected events in real-world manufacturing environments include resource-related events and operation-related events [1]. Disruption happens in various fields in supply chain and production environments. Supply disruption, production disruption, and transportation disruption are examples of disruption forms [2]. With the increase in SC activities and global business activities, the impact of disruption could be substantial [3]. Because of the uncertainty in disruption event occurrence and the seriousness of disruption impact, disruption handling is an important issue. Manufacturing systems should be flexible so as to absorb disturbance on a short horizon [1].

Flexible manufacturing systems (FMSs) produce a middle volume and a wide variety of part types [4,5]. The systems aim to achieve efficiency of mass production systems and flexibility of job shops. FMSs possess not only computer numerical control machines but also automated material handling devices. These devices include automated guided vehicles, rail-guided vehicles, robots, and so forth. Researchers categorize different types of FMSs mainly as flexible flow systems and general flexible machining systems [6]. It is very complicated to manage FMS production. An FMS with capacity constraint may not produce orders in time, resulting in some parts having to be sent to a job shop [7].

Supply chain engineering is a very important issue in the area of production research. Because of the serious impact of disruption in supply chains and in production systems, research efforts are placed on disruption handling in supply chains and in production systems. The mitigation of disruption risk can be made proactive or reactive. Therefore, there are two types of disruption handling approaches for production scheduling, that is, proactive scheduling and reactive scheduling [8]. Proactive scheduling takes into account unexpected disruption to build protection when schedules are generated. Reactive scheduling adjusts schedules when unexpected disruption events happen. Dynamic systems can be managed by applying advanced information technology such as Radio Frequency Identification (RFID). Therefore, the application of advanced information technology makes



Citation: He, Y.; Dolgui, A.; Smith, M. An Algorithm for Part Input Sequencing of Flexible Manufacturing Systems with Machine Disruption. *Algorithms* **2024**, *17*, 470. <https://doi.org/10.3390/a17100470>

Academic Editor: Roberto Montemanni

Received: 5 June 2024

Revised: 18 September 2024

Accepted: 24 September 2024

Published: 21 October 2024



Copyright: © 2024 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/>).

it possible to obtain and process information to handle disruption reactively. This paper applies reactive scheduling to handle machine disruption in FMS part input sequencing. The paper proposes an algorithm to provide a solution for part input sequencing of FMSs with machine breakage.

The remainder of the paper is described in the following. Section 2 presents related works. In Section 3, an algorithm for part input sequencing of FMSs to handle machine breakage is proposed. Evaluation with the analyses of the results of the proposed algorithm is described in Section 4. Conclusions, managerial implications, and further works are summarized finally in Section 5.

2. Related Works

Disruption handling in supply chain environments has been investigated by researchers. For example, selection of part suppliers and schedule of customer orders over a planning horizon were studied under disruption risk in supply chains with a solution approach proposed to optimize the expected cost and customer service level [9]. A mixed-integer programming (MIP)-based approach was developed for decision-making to simultaneously select part suppliers and schedule production and delivery in an SC with disruption risk [10]. The adjustment in order activity in a four-echelon SC for recovery from disruption was investigated with dynamic order-up-to policies developed to obtain the benefits of the dynamic policies incorporated by a metaheuristic parameter search [3]. A two-period modeling approach and a multi-period modeling approach with mixed-integer programming were developed with supply chain disruption risk, requiring a very short computational time to obtain proven optimal solutions for reasonably sized problems [11]. Production ordering dynamics in the situation of disruption were suggested after studying production ordering behavior in a supply chain under disruption risk [12]. Integration of lot sizing and supplier selection under disruption risk with lead time uncertainty was studied with reliability and the price of suppliers considered and polyhedral-budgeted uncertainty sets applied to obtain a lot size for minimizing total cost [13]. A novel quantitative approach was developed for SC viability under ripple effect with the two conflicting objectives of cost and customer service level considered to obtain very high computation efficiency [14].

Disruption handling in production systems has been studied. For example, the lot-sizing and sequencing problem was investigated for production lines considering random machine breakage with an optimal approach developed based on the decomposition of the problem [15]. A model and a solution approach were developed for production inventory management in an imperfect production environment with numerical examples demonstrated for real-time disruption recovery [16]. The continuous flow problem with processing capacity disruption was studied with schedule robustness considered and a method developed for schedule robustness analysis based on attainable sets [8]. A flexible production inventory model was proposed to manage production and inventory with the consideration of disruptions of demand and production, for a manufacturer to decrease losses [17]. A model was formulated by applying genetic algorithm as well as pattern search to handle production disruption for an imperfect production inventory system with multiple products and a single stage [18]. A heuristic-based column generation approach was proposed for production planning to mitigate disruption from demand uncertainty for flexible manufacturing systems with good numerical results [19].

Scheduling in flexible manufacturing environments with disruption has been investigated by researchers. The following provides a brief summary. In particular, flexible job shop (FJS) scheduling considering machine disruption is summarized. In early research, reactive scheduling policies were proposed, such as when-to-schedule, how-to-schedule, and so forth, for handling machine breakage and processing time variation in a flexible manufacturing system [20]. A genetic hybrid control architecture ORCA was proposed for an FMS, which could provide the ability to switch between a hierarchy and heterarchy architecture when an unexpected event occurs [21]. A game model was developed for the flexible job shop scheduling problem subject to machine breakdown with two objectives of

robustness and stability considered in their game procedure in rescheduling [22]. Evolutionary algorithms were applied to the FJS scheduling problem for improving makespan and stability with a comparison of the two proposed algorithms on example problems [23]. A hybrid approach was proposed for the FJS scheduling problem in dynamic environments for scheduling and rescheduling in disruption and experiments were performed for evaluation with the result of the competitiveness of the approach obtained [24]. Scheduling/rescheduling of flexible job shops were considered for machine recovery with an improved Jaya algorithm developed to minimize makespan in scheduling and to minimize both instability and makespan in rescheduling, generating the improved Jaya algorithm better than NSGAI and ISFLA in the non-dominated results [25]. Production scheduling and maintenance planning were considered in a flexible job shop in the situation of machine deterioration and a real-time system was proposed, which used a hybrid GA, an integrated model, and hybrid rescheduling policies [26]. A hybrid deep Q-network was built for dynamic FJS scheduling and for training to face disruption and experiments were performed to compare the method to scheduling rules, demonstrating the superiority of the proposed method [27]. A hierarchical-based deep reinforcement learning method was proposed for FJS scheduling and rescheduling and comparisons were made between the method and scheduling rules and other dynamic methods, demonstrating the superiority of the proposed method [28]. A flexible job shop scheduling method was proposed to consider machine breakdown and other dynamic events and to apply their dynamic event response strategy and their multi-objective model and to apply a multi-objective particle swarm arithmetic optimization [29]. Even though disruption handling in FJS scheduling and in FMS scheduling has been investigated by researchers, the investigation of disruption handling in FMS part input sequencing is not seen. This paper investigates disruption handling in FMS part input sequencing. An algorithm for part input sequencing of FMSs with machine breakage is proposed.

3. Proposed Algorithm

The application of segment set functions to FMS scheduling problems has been conducted. For example, the functions have been utilized for developing the simultaneous part input sequencing and robot scheduling algorithm to simultaneously sequence and input parts and to schedule a robot in FMSs [5]. The functions are utilized here in developing the proposed algorithm. The functions are described by discrete mathematics. Discrete mathematics involves algorithm, logic, Boolean algebras, and so forth [30].

Segment set functions include the concepts of sets, domains, ranges, parts, and so forth. The functions consist of pair-wise elements of domains and ranges. For a simple set function, the 1st element of the function is a part in a set. The 2nd element of the function is an integer number representing the range of the function. For a transform function, the 1st element of the function is also a part in a set. The 2nd element of the function is the range of the function. The domain of the function is in multiple regions. The range of the function has segment values corresponding to different sets of parts. For a weight function, the 1st element of the function is similarly a part in a set. The 2nd element of the function is the range of the function. Different weights are assigned to the function to correspond to different sets of parts. For an overall function, the 1st element of the function is similarly a part in a set. The 2nd element of the function is similarly the range of the function. The range of the function has segment values corresponding to different sets.

A set of parts in the preprocess area of an FMS at time t are denoted as $A_x(t)$, $A_x(t) = \{b_{hi} | g_{b_{hi}x}(t) = 1\}$, where x is a part set indicator, b_{hi} is part h of order i , i is an order index, $i = 1, 2, \dots$, h is a part index, $h = 1, 2, \dots, r_i$, r_i is the production requirement for order i ; $g_{b_{hi}x}(t)$ is the part set status, $g_{b_{hi}x}(t) = \begin{cases} 1, & \text{part } b_{hi} \text{ is in set } A_x(t); \\ 0, & \text{otherwise.} \end{cases}$

$A_x(t)$ is classified as the other two sets of $A_a(t)$ and $A_b(t)$. Subset $A_a(t)$ is a balanced set. Subset $A_b(t)$ is an unbalanced set.

$$A_a(t) = \{b_{hi} | b_{hi} \in A_x(t) \wedge g_{b_{hi}a}(t) = 1 \wedge M_{b_{hi}1} = \hat{j}\}, \quad (1)$$

$$A_b(t) = \{b_{hi} | b_{hi} \in A_x(t) \wedge g_{b_{hi}b}(t) = 1 \wedge M_{b_{hi}1} \neq \hat{j}\}, \quad (2)$$

$A_a(t)$ and $A_b(t)$ are classified as other four sets, $A_u(t)$, $A_v(t)$, $A_m(t)$, and $A_n(t)$. Among those, $A_u(t)$ and $A_v(t)$ are subsets of $A_a(t)$.

$$A_u(t) = \{b_{hi} | b_{hi} \in A_a(t) \wedge g_{b_{hi}u}(t) = 1 \wedge M_{b_{hi}1} = \hat{j} \wedge M_{b_{hi}2} = \tilde{j}\}, \quad (3)$$

$$A_v(t) = \{b_{hi} | b_{hi} \in A_a(t) \wedge g_{b_{hi}v}(t) = 1 \wedge M_{b_{hi}1} = \hat{j} \wedge M_{b_{hi}2} \neq \tilde{j}\}, \quad (4)$$

$A_m(t)$ and $A_n(t)$ are subsets of $A_b(t)$.

$$A_m(t) = \{b_{hi} | b_{hi} \in A_b(t) \wedge g_{b_{hi}m}(t) = 1 \wedge M_{b_{hi}1} \neq \hat{j} \wedge M_{b_{hi}1} = \tilde{j}\}. \quad (5)$$

$$A_n(t) = \{b_{hi} | b_{hi} \in A_b(t) \wedge g_{b_{hi}n}(t) = 1 \wedge M_{b_{hi}1} \neq \hat{j} \wedge M_{b_{hi}1} \neq \tilde{j}\}. \quad (6)$$

The symbols in the above equations are explained in the following. $g_{b_{hi}q}(t)$ indicates the status of a part set, $g_{b_{hi}q}(t) = \begin{cases} 1, & \text{part } b_{hi} \text{ is in set } A_q(t); \\ 0, & \text{otherwise.} \end{cases}$ $M_{b_{hi}k}$ expresses the machine for k of part b_{hi} , k is an operation index. \hat{j} expresses the machine that has the minimum of $\eta_j(t)$. \tilde{j} expresses the machine that has the second minimum of $\eta_j(t)$. j is a machine indicator. $\eta_j(t)$ expresses the workload of machine j . For the above sets, the segment set functions can be obtained. A detailed description of these segment set functions is provided in [5].

An algorithm is proposed by applying the segment set functions to part input sequencing of FMSs for handling machine breakage. The proposed algorithm is segment set-based. It also applies the earliest due date scheduling rule for machine scheduling. Its aim is to achieve part input sequencing of FMSs to handle a machine breakage. It is simply called the machine disruption handling algorithm (MDH Algorithm, Algorithm 1). The proposed algorithm is depicted as follows. Additional symbols utilized in the algorithm are listed in Table 1.

Algorithm 1: Machine Disruption Handling Algorithm

- Step 1. Initialize $t = t_0, \rho(t) = 0, \delta(t) = 0, m_r(t) = 0, m_j(t) = 1, j \in J$.
 - Step 2. Check the current t , If $t \geq T$, Stop.
 - Step 3. Check the current status of part b_{hi} . If part b_{hi} finishes processing, $\delta(t) = 1$, then $t = c_{hi}$. If $\delta(t) = 0$, go to Step 10.
 - Step 4. Obtain machine status. $m_j(t), j \in J$. If $\forall j \in J, m_j(t) \vee m_r(t) = 0$, then identify and remove broken machine. Remove parts from broken machine.
 $BM(t) = \{j | j \in J \vee m_j(t) = 0\}, t = t_p, m_r(t) = 1$.
 - Step 5. Place parts in preprocess area in $A_x(t)$, $A_x(t) = \{b_{hi} | g_{b_{hi}x}(t) = 1\}$. Update parts at $BM(t)$ in set $A_e(t)$, $A_e(t) = \{b_{hi} | g_{b_{hi}e}(t) = 1\}$. Update parts in $A_x(t)$ for not processing at $BM(t)$ in set $A_o(t)$, $A_o(t) = \{b_{hi} | b_{hi} \in A_x(t) \wedge g_{b_{hi}o}(t) = 1\}$. Update parts in set $A_x(t)$ for processing at $BM(t)$ in set $A_c(t)$, $A_c(t) = \{b_{hi} | b_{hi} \in A_x(t) \wedge g_{b_{hi}c}(t) = 1\}$.
 - Step 6. If $(m_r(t) \neq 0) \wedge (A_o(t) \neq \phi)$, then $A_y(t) = \{b_{hi} | b_{hi} \in A_o(t)\}$, go to **Step 7**, else if $(m_r(t) = 0) \wedge (A_e(t) \neq \phi)$, then $A_y(t) = \{b_{hi} | b_{hi} \in A_e(t)\}$, go to **Step 7**, else if $(m_r(t) = 0) \wedge (A_c(t) \neq \phi) \vee (m_r(t) = 0) \wedge (A_o(t) \neq \phi)$, then $A_y(t) = \{b_{hi} | \{b_{hi} \in A_c(t)\} \cup \{b_{hi} \in A_o(t)\}\}$.
-

Algorithm 1. *Cont.*

Step 7. Obtain workload in the FMS at time t . Obtain the least workload machine,

$$\tilde{j} = \left\{ j \mid j \in J \wedge \eta_j(t) = \min_{j \in J} \eta_j(t) \right\}. \text{ Also, obtain the 2nd least workload machine}$$

$$\tilde{j} = \left\{ j \mid j \in J \wedge \eta_j(t) = \min_{j \in J, j \neq \tilde{j}} \eta_j(t) \right\}. A_x(t) = A_y(t). \text{ Apply Equation (1) to (6) to obtain subsets}$$

$A_q(t)$ for $q = a, b$, also for $q = u, v, m, n$. $g_{b_{hi}q}(t) = 1$, for $q = x, a, b$, also for $q = u, v, m, n$.

Step 8. Obtain segment set functions by equations in [5]. Obtain the simple set functions λ_q of set set $A_q(t)$ by Equation (13) for $q = x, a, b$, also for $q = u, v, m, n$. Obtain the transform function $\bar{\lambda}$ by Equations (14) and (19). Assign weights ζ_q for $q = u, v, m, n$. $\zeta_u = -5K$ for $A_u(t)$, $\zeta_v = -3K$ for $A_v(t)$, $\zeta_m = -K$ for $A_m(t)$, $\zeta_n = 0$ for $A_n(t)$. $K = 0.5S$. Obtain the weight function $\bar{\lambda}$ applying Equations (15) and (18). Obtain the overall function $\hat{\lambda}$ by Equations (16), (17) and (20).

Step 9. Obtain the minimal value of $\hat{\lambda}$, $\hat{\lambda}(t) = \min_{b_{hi} \in A_x(t)} \{ \hat{\lambda}_q(t, b_{hi}), q = u, v, m, n \}$. Obtain the

input part

$$b^* = \left\{ b_{hi} \mid b_{hi} \in A_x(t) \wedge a_{hi} = \min_{b_{hi} \in A_x(t)} \{ a_{hi} \mid \hat{\lambda}_q(t, b_{hi}) = \hat{\lambda}(t), q = u, v, m, n \} \right\}, \delta(t) = 0. u(t) = m_{b_{hi}1} + t, t = u(t).$$

Step 10. If $\rho(t) = 1$, $u(t) = f_{b_{hi}k}(t)$, $t = u(t)$. Obtain $m_{b_{hi}k}(t)$, $g = m_{b_{hi}k}(t)$, $\rho(t) = 0$, Obtain machine queue set $A_g(t) = \{ b_{hi} \mid g_{b_{hi}g}(t) = 1 \}$, else go to Step 2.

Step 11. Identify the part to be processed in the following.

$$p^* = \left\{ b_{hi} \mid b_{hi} \in A_g(t) \wedge a_{hi} = \min_{b_{hi} \in A_g(t)} \left\{ a_{hi} \mid d_{hi} = \min_{b_{hi} \in A_g(t)} d_{hi} \right\} \right\}, g_{b_{hi}g}(t) = 0, \text{ go to Step 2.}$$

Table 1. Additional symbols utilized in Algorithm 1.

Notation	Explanation
Indices and Sets	
g	Machine queue set indicator
J	Machine set, $J = \{1, 2, \dots, M\}$
q	Part set indicator
Parameters	
a_{hi}	Arrival time of b_{hi}
d_{hi}	Due date of b_{hi}
K	Constant
M	Number of machines
$m_{b_{hi}k}$	Robot move time for k of b_{hi}
S	Size of preprocess area
T	Production cycle
t_0	Initial time
ζ_q	Weight of $A_q(t)$
Variables	
$A_c(t)$	Part set for processing at $BM(t)$
$A_e(t)$	Part set needs repairing at t
$A_g(t)$	Machine queue set g at t
$A_o(t)$	Part set for not processing at $BM(t)$
$A_q(t)$	Part set q at t
$A_y(t)$	Part set y at t
b^*	Part for inputting
$BM(t)$	Broken machine at t
c_{hi}	Completion time of b_{hi}
$f_{b_{hi}k}(t)$	Completion time of k of b_{hi} at t
$g_{b_{hi}g}(t)$	$g_{b_{hi}g}(t) = \begin{cases} 1, & b_{hi} \text{ is in } A_g(t); \\ 0, & \text{otherwise.} \end{cases}$
$g_{b_{hi}q}(t)$	$g_{b_{hi}q}(t) = \begin{cases} 1, & b_{hi} \text{ is in } A_q(t); \\ 0, & \text{otherwise.} \end{cases}$
$m_{b_{hi}k}(t)$	Machine finishing k of b_{hi} at t

Table 1. *Cont.*

Notation	Explanation
$m_j(t)$	Machine operating status if it is available at t $m_j(t) = \begin{cases} 1, & \text{machine } j \text{ is available at } t; \\ 0, & \text{otherwise.} \end{cases}$
$m_r(t)$	Machine repair status if a broken machine is repairing at t $m_r(t) = \begin{cases} 1, & \text{a broken machine is repairing at } t; \\ 0, & \text{otherwise.} \end{cases}$
p^*	Part for processing
t_p	Machine broken time
$\delta(t)$	Part processing status when a part finishes its processing at t $\delta(t) = \begin{cases} 1, & \text{a part finishes its processing at } t; \\ 0, & \text{otherwise.} \end{cases}$
$\bar{\lambda}$	Transform function
$\tilde{\lambda}$	Weight function
$\hat{\lambda}$	Overall function
λ_q	Simple set function for $A_q(t)$
$\hat{\lambda}(t)$	Minimal value of $\hat{\lambda}$
$\hat{\lambda}(t, b_{hi})$	Range of $\hat{\lambda}$ for b_{hi} at t
$\mu(t)$	Temporary completion time
$\rho(t)$	Part operation status when a part finishes an operation at t $\rho(t) = \begin{cases} 1, & \text{a part finishes an operation at } t; \\ 0, & \text{otherwise.} \end{cases}$

The proposed MDH Algorithm (Algorithm 1) is a dynamic algorithm. It utilizes the information of the dynamic workload to make an input decision. The dynamic workload is described in [5]. When a part completes operations, Algorithm 1 then inputs a part dynamically. The proposed algorithm identifies a broken machine on an FMS shop floor. It also identifies part processing at a broken machine. The algorithm handles machine breakage according to a different part processing status at a broken machine to identify parts for inputting. Advanced information technology like RFID can be applied for the implementation of a shop floor monitoring system. The shop floor monitoring system collects and processes dynamic information on an FMS shop floor. The proposed algorithm runs with the shop floor monitoring system that applies RFID to identify and handle machine breakage.

4. Evaluation with Result Analyses

The evaluation of the proposed algorithm is based on a simulation. It is difficult to simulate machine breakage and repair in flexible manufacturing systems in real-world production environments. Therefore, the proposed Algorithm 1 is evaluated by simulation experiments and statistical analyses in the situation with no machine breakage. The proposed algorithm is compared to an FMS part input sequencing algorithm, the state-dependent part input algorithm (SPI algorithm) in the literature [31]. The compared algorithm, the SPI algorithm (Algorithm A1), is provided in Appendix A.

The simulation model of the FMS and the simulation experiment settings are the same as those in [5,31]. One of the FMS scenarios in the numerical study in [31] does not obtain the best or the worst results among the four scenarios studied. This scenario was used for numerical study in [5]. This scenario is also utilized here for evaluating the proposed algorithm. The data used for evaluation are provided in Appendix B. Due dates for parts are $d_{hi} = 7500 + U(0, 6500)$. The adjustable constant is 7500 s. The uniformly distributed random variable is in the range of 0 to 6500 s. The parameters are set according to preliminary experiments so that the FMS generates approximately thirty percent of tardy parts.

Two approaches are compared using common random number technique for each pair of approaches so as to decrease variance. Ten independent simulation runs are performed with terminating simulation used. The simulation time per run for each approach is 200,000 s or 3333 min. There are more than 1000 parts produced during this simulation time. The system is in a steady state.

The performance measures used for evaluating the proposed algorithm include TP, MF, and RU. TP represents total parts produced, that is, the total number of parts produced in a production cycle. MF represents mean flowtime, that is, the total flowtime divided by the total parts produced. RU represents robot utilization, that is, the sum of the total time of robot moves divided by a production cycle.

The simulation data of TP, MF, and RU are analyzed. Averages of the performance measures TP, MF, and RU of Algorithm 1 from 10 independent simulation runs are displayed in Table 2. Averages of the performance measures TP, MF, and RU of Algorithm A1 from 10 independent simulation runs are also displayed in Table 2. The absolute improvement of Algorithm 1 versus Algorithm A1 is computed. The relative improvement of Algorithm 1 versus Algorithm A1 is also computed. The following equations are utilized for computing the absolute improvement and the relative improvement.

$$\omega = \left(\sum_{r=1}^{10} (\psi_r - \varphi_r) \right) / 10, \tag{7}$$

$$\omega = \left(\sum_{r=1}^{10} (\psi_r - \varphi_r) / \varphi_r \right) * 10, \tag{8}$$

where ω is absolute improvement. ω is relative improvement (%). r is simulation run index. ψ_r is the performance measure of approach ψ in simulation run r . φ_r is the performance measure of approach φ in simulation run r . The absolute improvement and relative improvement of Algorithm 1 versus Algorithm A1 for all performance measures TP, MF, and RU are also in Table 2.

Table 2. Comparison of Algorithm 1 to Algorithm A1.

Measure	TP (Parts)	MF (Minutes)	RU (%)
Algorithm 1	1254.9	86.38	71.03
Algorithm A1	1251.8	87.36	70.87
ω	3.1	0.98	0.16
ω (%)	0.25	1.12	0.23
Test statistic	1.4 *	2.44 *	1.17

Note: A bold number indicates the improvement of a performance measure. * indicates significant improvement of a performance measure.

It can be seen from the table that the averages of TP, MF, and RU by the proposed Algorithm 1 are 1254.9 parts, 86.4 min, and 71.03%, respectively. The averages of TP, MF, and RU by the comparative Algorithm A1 are 1251.8 parts, 87.4 min, and 70.87%, respectively. Algorithm 1 has better performance than Algorithm A1 for all performance measures of TP, MF, and RU as shown in the table. In the table, the absolute improvements ω of TP, MF, and RU by Algorithm 1 versus Algorithm A1 are 3.1 parts, 0.98 min, and 0.16%, respectively. The relative improvements ω of TP, MF, and RU for Algorithm 1 versus Algorithm A1 are 0.25 parts, 1.12 min, and 0.23%, respectively. The absolute and relative improvements in the table display the improvements of all the performance measures: TP, MF, and RU. The values in the table display that all performance measures obtained by Algorithm 1 are better than those obtained by Algorithm A1.

Significance tests are applied. The paired t -tests are conducted. The significance level is 0.1. The t -test statistic has the critical value of 1.37 at a significance level of 0.1. The test statistics obtained for the relative improvements by Algorithm 1 are 1.4, 2.44, and 1.17 for TP, MF, and RU, respectively, as illustrated in Table 2. The results show that TP and MF are improved significantly. Because MF improves, parts are produced faster with more parts

produced. That TP and MF are significantly improved indicates a significant production increase. The results indicate that production is significantly increased by Algorithm 1 in comparison to Algorithm A1.

In summary, the comparative results show significant production increase by Algorithm 1 compared to Algorithm A1. All performance measures of TP, MF, and RU obtained by Algorithm 1 show improvements in comparison to the comparative Algorithm A1. The comparative results indicate that the performance of Algorithm 1 is improved compared to Algorithm A1. That is, Algorithm 1 is superior to Algorithm A1 from the literature in the situation with no machine breakage.

5. Conclusions and Future Work

Disruption happens in the real world in supply chain and production environments. This paper studies disruption handling of machine breakage in FMS part input sequencing. The MDH Algorithm is proposed for part input sequencing of FMSs with machine breakage. The proposed algorithm is based on reactive scheduling. Because of the difficulty in simulating FMS machine breakage and repair in real-world production environments, the proposed algorithm is evaluated in a situation with no machine breakage. The proposed algorithm is compared to an existing FMS part input sequencing algorithm from the literature, the state-dependent part input algorithm. The comparative results indicate that the proposed MDH Algorithm improves the performance significantly, generating the significant increase in total parts produced and mean flowtime decrease in the situation with no machine breakage. The evaluation results indicate the superiority of the MDH Algorithm in comparison to the state-dependent part input algorithm in terms of total parts produced, mean flowtime, and robot utilization in the situation with no machine breakage.

This paper contributes an applicable and effective algorithm for part input sequencing of FMSs to handle machine breakage. Managerial implications include the following. The proposed algorithm provides an applicable approach to the managers of FMSs to make FMS part input sequencing decisions for handling machine breakage. Disruption usually happens unpredictably in the real world. Real-time decision making applying advanced information technology such as RFID makes it possible to detect and handle disruption reactively and quickly. The proposed algorithm makes it possible to realize real-time decision making for part input sequencing of FMSs with machine breakage.

There are more random factors that affect FMS part input sequencing such as high-tech devices added on an FMS shop floor and rushed orders arriving at an FMS. Suggestions for future research could be to develop more effective algorithms to handle more situations of disruption in FMS part input sequencing. Additional suggestions for future research could be the development of decision support systems for part input sequencing of FMSs to handle machine disruption.

Author Contributions: Conceptualization, Y.H. and A.D.; formal analysis, Y.H.; investigation, Y.H.; methodology, Y.H., A.D. and M.S.; writing—original draft, Y.H.; writing—review and editing, Y.H., A.D. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: The second author's research was funded by the French National Research Agency (ANR), project ANR-21-CE10-0019.

Data Availability Statement: Data used in the performance evaluation in this study are from the previous studies. They are cited and are also provided in the Appendices A and B in this paper. Additional supportive data can be made available from Yumin He upon reasonable request.

Acknowledgments: The efforts of the Academic Editor and the reviewers are appreciated. The comments and suggestions helped to improve our paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Algorithm A1: State-Dependent Part Input Algorithm

Step 1. Form part set $M(t)$ from waiting parts in the preprocess area of an FMS at time t .

Step 2. Partition the parts in $M(t)$ into subsets of balanced set $X(t)$ and unbalanced set $Y(t)$. $X(t)$ possesses the parts having their first operation at the least loaded machine to help balance workload. $Y(t)$ possesses the parts not having the first operation at the least loaded machine. $X(t)$ and $Y(t)$ are further divided into another two subsets individually so that $M(t) = \cup_q G_q(t), q = \beta, \gamma, \mu, \nu$. $X(t) = G_\beta(t) \cup G_\gamma(t)$. $Y(t) = G_\mu(t) \cup G_\nu(t)$.

Step 3. Obtain the following simple set functions, λ_e for $M(t)$, λ_s for balanced set, and λ_p for unbalanced set,
 $\lambda_e : M(t) \rightarrow I. \lambda_e = \{ (\alpha_i, \lambda_e(t, \alpha_i)) \mid \alpha_i \in M(t), \lambda_e(t, \alpha_i) \in I \}$. $\lambda_s : X(t) \rightarrow I. \lambda_s = \{ (\alpha_i, \lambda_s(t, \alpha_i)) \mid \alpha_i \in X(t), \lambda_s(t, \alpha_i) \in I \}$. $\lambda_p : Y(t) \rightarrow I. \lambda_p = \{ (\alpha_i, \lambda_p(t, \alpha_i)) \mid \alpha_i \in Y(t), \lambda_p(t, \alpha_i) \in I \}$. Obtain the simple set functions $\lambda_q, q = \beta, \gamma, \mu, \nu$ for the subsets of balanced and unbalanced sets $G_q(t), q = \beta, \gamma, \mu, \nu$. $\lambda_\beta : G_\beta(t) \rightarrow I. \lambda_\beta = \{ (\alpha_i, \lambda_\beta(t, \alpha_i)) \mid \alpha_i \in G_\beta(t), \lambda_\beta(t, \alpha_i) \in I \}$. $\lambda_\gamma : G_\gamma(t) \rightarrow I. \lambda_\gamma = \{ (\alpha_i, \lambda_\gamma(t, \alpha_i)) \mid \alpha_i \in G_\gamma(t), \lambda_\gamma(t, \alpha_i) \in I \}$. $\lambda_\mu : G_\mu(t) \rightarrow I. \lambda_\mu = \{ (\alpha_i, \lambda_\mu(t, \alpha_i)) \mid \alpha_i \in G_\mu(t), \lambda_\mu(t, \alpha_i) \in I \}$. $\lambda_\nu : G_\nu(t) \rightarrow I. \lambda_\nu = \{ (\alpha_i, \lambda_\nu(t, \alpha_i)) \mid \alpha_i \in G_\nu(t), \lambda_\nu(t, \alpha_i) \in I \}$.

Step 4. Obtain the segment set function, $\lambda' : M(t) \rightarrow I$.
 $\lambda' = \{ \{ (\alpha_i, \lambda_q(t, \alpha_i)) \mid \alpha_i \in G_q(t), \lambda_q(t, \alpha_i) \in I \}, q = \beta, \gamma, \mu, \nu \}$. Obtain the transform function that is also a segment set function, $\bar{\lambda} : M(t) \rightarrow I. \bar{\lambda} = \{ \{ (\alpha_i, \bar{\lambda}_q(t, \alpha_i)) \mid \alpha_i \in G_q(t), \bar{\lambda}_q(t, \alpha_i) \in I \}, q = \beta, \gamma, \mu, \nu \}$. $\bar{\lambda}_\beta(t, \alpha_i) = \lambda_s(t, \alpha_i) + \lambda_\beta(t, \alpha_i), \alpha_i \in G_\beta(t)$; $\bar{\lambda}_\gamma(t, \alpha_i) = \lambda_s(t, \alpha_i), \alpha_i \in G_\gamma(t)$; $\bar{\lambda}_\mu(t, \alpha_i) = \lambda_\mu(t, \alpha_i), \alpha_i \in G_\mu(t)$; $\bar{\lambda}_\nu(t, \alpha_i) = 0, \alpha_i \in G_\nu(t)$.

Step 5. Assign weights $\zeta_q, q = \beta, \gamma, \mu, \nu, \zeta_\beta = -5K$ for $G_\beta(t), \zeta_\gamma = -3K$ for $G_\gamma(t), \zeta_\mu = -K$ for $G_\mu(t), \zeta_\nu = 0$ for $G_\nu(t)$. $K = 0.5S. \hat{\lambda}_s(t, \alpha_i) < \hat{\lambda}_p(t, \alpha_i)$ is satisfied as explained in [31].

Step 6. Obtain the weight function,
 $\tilde{\lambda} : M(t) \rightarrow I. \tilde{\lambda} = \{ \{ (\alpha_i, \tilde{\lambda}_q(t, \alpha_i)) \mid \alpha_i \in G_q(t), \tilde{\lambda}_q(t, \alpha_i) \in I \}, q = \beta, \gamma, \mu, \nu \}$. $\tilde{\lambda}_q(t, \alpha_i) = \bar{\lambda}_q(t, \alpha_i) + \zeta_q, q = \beta, \gamma, \mu, \nu$. Obtain the overall function,
 $\hat{\lambda} : M(t) \rightarrow I. \hat{\lambda} = \{ \{ (\alpha_i, \hat{\lambda}_q(t, \alpha_i)) \mid \alpha_i \in G_q(t), \hat{\lambda}_q(t, \alpha_i) \in I \}, q = \beta, \gamma, \mu, \nu \}$. $\hat{\lambda}_\beta(t, \alpha_i) = \lambda_e(t, \alpha_i) + \lambda_s(t, \alpha_i) + \lambda_\beta(t, \alpha_i) + \zeta_\beta, \alpha_i \in G_\beta(t)$; $\hat{\lambda}_\gamma(t, \alpha_i) = \lambda_e(t, \alpha_i) + \lambda_s(t, \alpha_i) + \zeta_\gamma, \alpha_i \in G_\gamma(t)$; $\hat{\lambda}_\mu(t, \alpha_i) = \lambda_e(t, \alpha_i) + \lambda_\mu(t, \alpha_i) + \zeta_\mu, \alpha_i \in G_\mu(t)$; $\hat{\lambda}_\nu(t, \alpha_i) = \lambda_e(t, \alpha_i) + \zeta_\nu, \alpha_i \in G_\nu(t)$.

Step 7. Identify the minimal value of $\hat{\lambda}, \hat{\lambda}(t) = \min_{\alpha_i \in M(t)} \{ \hat{\lambda}_q(t, \alpha_i), q = \beta, \gamma, \mu, \nu \}$. Then, input the part corresponding to the minimal element in the range of the overall function and arriving the earliest.

Table A1. Symbols in state-dependent part input algorithm.

Notation	Explanation
	Indices
α_i	Part in order $i, i = 1, 2, \dots$
q	Part set indicator
	Parameters
K	Constant
S	Size of preprocess area
ζ_q	Weight of $G_q(t)$
	Variables
$G_q(t)$	Part set q at t
$M(t)$	Set of parts in preprocess area at t
$X(t)$	Balanced set of parts at t
$Y(t)$	Unbalanced set of parts at t
λ_e	Simple set function for $M(t)$
λ_p	Simple set function for $Y(t)$
λ_q	Simple set function for $G_q(t)$
λ_s	Simple set function for $X(t)$
λ'	Segment set function
$\bar{\lambda}$	Transform function

Table A1. Cont.

Notation	Explanation
$\tilde{\lambda}$	Weight function
$\hat{\lambda}$	Overall function
$\hat{\lambda}(t)$	Minimal value of $\hat{\lambda}$
$\lambda_e(t, \alpha_i)$	Range of λ_e for $M(t)$ at t
$\lambda_p(t, \alpha_i)$	Range of λ_p for $Y(t)$ at t
$\lambda_q(t, \alpha_i)$	Range of λ_q for $G_q(t)$ at t
$\bar{\lambda}_q(t, \alpha_i)$	Range of $\bar{\lambda}$ for $G_q(t)$ at t
$\tilde{\lambda}_q(t, \alpha_i)$	Range of $\tilde{\lambda}$ for $G_q(t)$ at t
$\hat{\lambda}_q(t, \alpha_i)$	Range of $\hat{\lambda}$ for $G_q(t)$ at t
$\lambda_s(t, \alpha_i)$	Range of λ_s for $X(t)$ at t

Appendix B Data Used in Evaluation

Table A2. Production information.

Part Type	Prod. Req.	Part Type	Prod. Req.	Part Type	Prod. Req.
1	6%	10	8%	19	6%
2	2%	11	2%	20	4%
3	2%	12	6%	21	4%
4	2%	13	2%	22	6%
5	10%	14	4%	23	4%
6	2%	15	6%	24	2%
7	6%	16	4%	25	4%
8	2%	17	2%		
9	2%	18	2%		

Table A3. Processing times and part routes.

Type	Route	Processing Time (Seconds)
1	1 6 3 8	115 165 135 285
2	6 8 9 4 7 1 2 5 3	165 185 195 145 175 115 125 155 135
3	4 1 2 5 6	45 15 25 55 65
4	5 1 4 2 6 8	155 115 145 125 165 185
5	4 2 6 8 1	245 225 265 185 15
6	3 1 8	35 15 85
7	6 7 4 5 1 3 2	65 175 45 155 115 135 125
8	8 4 2	185 345 325
9	2 5 6 3 1 4	225 255 165 135 115 245
10	9 1 7 8	195 115 175 185
11	4 5 3 1 9 7 2 8 6	245 255 35 115 295 275 225 185 165
12	8 5 6 3 9 1 7	185 155 65 135 195 15 175
13	3 2 6 7 5	135 125 165 175 155
14	5 1 4 7 6 2	450 110 440 470 160 420
15	7 1 3 5	70 210 130 250
16	6 4 1 8 2	160 140 110 180 120
17	3 5 2 8 1	130 350 320 180 110
18	6 3 1 8 4	60 30 10 80 40
19	1 7 6 8 2 4 3	110 470 160 180 420 440 130
20	9 6 5 1 8 4 2	195 165 155 15 185 145 125
21	2 7 5 3	225 275 255 135
22	7 5 6 2 3 1	170 150 160 120 130 110
23	4 1 7 2 5	145 115 175 125 155
24	7 4 1 9	75 45 15 95
25	8 3 5 2 6 4 1 9	85 235 255 225 165 245 115 295

References

1. Proth, J.M. Scheduling: New trends in industrial environment. *Annu. Rev. Control* **2007**, *31*, 157–166. [\[CrossRef\]](#)
2. Ivanov, D.; Dolgui, A.; Sokolov, B.; Ivanova, M. Literature review on disruption recovery in the supply chain. *Int. J. Prod. Res.* **2017**, *55*, 6158–6174. [\[CrossRef\]](#)
3. Schmitt, T.G.; Kumar, S.; Stecke, K.E.; Glover, F.W.; Ehlen, M.A. Mitigating disruptions in a multi-echelon supply chain using adaptive ordering. *Omega* **2017**, *68*, 185–198. [\[CrossRef\]](#)
4. Stecke, K.E. Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. *Manag. Sci.* **1983**, *29*, 273–288. [\[CrossRef\]](#)
5. He, Y.; Stecke, K.E. Simultaneous part Input sequencing and robot scheduling for mass customisation. *Int. J. Prod. Res.* **2022**, *60*, 2481–2496. [\[CrossRef\]](#)
6. Rachamadugu, R.; Stecke, K.E. Classification and review of FMS scheduling procedures. *Prod. Plan. Control* **1994**, *5*, 2–20. [\[CrossRef\]](#)
7. Tetzlaff, U.A.W.; Pesch, E. Optimal workload allocation between a job shop and an FMS. *IEEE Trans. Robot. Autom.* **1999**, *15*, 20–32. [\[CrossRef\]](#)
8. Ivanov, D.; Dolgui, A.; Sokolov, B.; Werner, F. Schedule robustness analysis with the help of attainable sets in continuous flow problem under capacity disruptions. *Int. J. Prod. Res.* **2016**, *54*, 3397–3413. [\[CrossRef\]](#)
9. Sawik, T. On the fair optimization of cost and customer service level in a supply chain under disruption risks. *Omega* **2015**, *53*, 58–66. [\[CrossRef\]](#)
10. Sawik, T. Integrated supply, production and distribution scheduling under disruption risks. *Omega* **2016**, *62*, 131–144. [\[CrossRef\]](#)
11. Sawik, T. Two-period vs. multi-period model for supply chain disruption management. *Int. J. Prod. Res.* **2019**, *57*, 4502–4518. [\[CrossRef\]](#)
12. Ivanov, D. Disruption tails and revival policies: A simulation analysis of supply chain design and production-ordering systems in the recovery and post-disruption periods. *Comput. Ind. Eng.* **2019**, *127*, 558–570. [\[CrossRef\]](#)
13. Thevenin, S.; Ben-Ammar, O.; Brahimi, N. Robust optimization approaches for purchase planning with supplier selection under lead time uncertainty. *Eur. J. Oper. Res.* **2022**, *303*, 1199–1215. [\[CrossRef\]](#)
14. Sawik, T. A stochastic optimisation approach to maintain supply chain viability under the ripple effect. *Int. J. Prod. Res.* **2023**, *61*, 2452–2469. [\[CrossRef\]](#)
15. Dolgui, A.; Levin, G.; Louly, M.-A. Decomposition approach for a problem of lot-sizing and sequencing under uncertainties. *Int. J. Comput. Integr. Manuf.* **2005**, *18*, 376–385. [\[CrossRef\]](#)
16. Paul, S.K.; Sarker, R.; Essam, D. Managing disruption in an imperfect production–inventory system. *Comput. Ind. Eng.* **2015**, *84*, 101–112. [\[CrossRef\]](#)
17. Xu, X.; Shang, J.; Wang, H.; Chiang, W.-C. Optimal production and inventory decisions under demand and production disruptions. *Int. J. Prod. Res.* **2016**, *54*, 287–301. [\[CrossRef\]](#)
18. Malik, A.I.; Sarkar, B. Disruption management in a constrained multi-product imperfect production system. *J. Manuf. Syst.* **2020**, *56*, 227–240. [\[CrossRef\]](#)
19. Elyasi, M.; Altan, B.; Ekici, A.; Özener, O.O.; Yanıkoğlu, I.; Dolgui, A. Production planning with flexible manufacturing systems under demand uncertainty. *Int. J. Prod. Res.* **2024**, *62*, 157–170. [\[CrossRef\]](#)
20. Sabuncuoğlu, I.; Kizilishik, O.B. Reactive scheduling in a dynamic and stochastic FMS environment. *Int. J. Prod. Res.* **2003**, *41*, 4211–4231. [\[CrossRef\]](#)
21. Pach, C.; Berger, T.; Bonte, T.; Trentesaux, D. ORCA-FMS: A dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. *Comput. Ind.* **2014**, *65*, 706–720. [\[CrossRef\]](#)
22. Sun, D.; He, W.; Zheng, L.; Liao, X. Scheduling flexible job shop problem subject to machine breakdown with game theory. *Int. J. Prod. Res.* **2014**, *52*, 3858–3876. [\[CrossRef\]](#)
23. Ahmadi, E.; Zandieh, M.; Farrokh, M.; Emami, S.M. A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. *Comput. Oper. Res.* **2016**, *73*, 56–66. [\[CrossRef\]](#)
24. Zhang, S.; Wong, T.N. Flexible job-shop scheduling/rescheduling in dynamic environment: A hybrid MAS/ACO approach. *Int. J. Prod. Res.* **2017**, *55*, 3173–3196. [\[CrossRef\]](#)
25. Gao, K.; Yang, F.; Li, J.; Sang, H.; Luo, J. Improved Jaya algorithm for flexible job shop rescheduling problem. *IEEE Access* **2020**, *8*, 86915–86922. [\[CrossRef\]](#)
26. Ghaleb, M.; Taghipour, S.; Zolfagharinia, H. Real-time integrated production-scheduling and maintenance-planning in a flexible job shop with machine deterioration and condition-based maintenance. *J. Manuf. Syst.* **2021**, *61*, 423–449. [\[CrossRef\]](#)
27. Li, Y.; Gu, W.; Yuan, M.; Tang, Y. Real-time data-driven dynamic scheduling for flexible job shop with insufficient transportation resources using hybrid deep Q network. *Robot. Comput. Integr. Manuf.* **2022**, *74*, 102283. [\[CrossRef\]](#)
28. Luo, S.; Zhang, L.; Fan, Y. Real-time scheduling for dynamic partial-no-wait multiobjective flexible job shop by deep reinforcement learning. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 3020–3038. [\[CrossRef\]](#)
29. Duan, J.; Wang, J. Robust scheduling for flexible machining job shop subject to machine breakdowns and new job arrivals considering system reusability and task recurrence. *Expert Syst. Appl.* **2022**, *203*, 117489. [\[CrossRef\]](#)

30. Johnsonbaugh, R. *Discrete Mathematics*; Prentice Hall: Hoboken, NJ, USA, 2004; pp. 55–111.
31. He, Y.; Rachamadugu, R.; Smith, M.L.; Stecke, K.E. Segment set-based part input sequencing in flexible manufacturing systems. *Int. J. Prod. Res.* **2015**, *53*, 5106–5117. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.