Journal of Manufacturing Systems 80 (2025) 824–840

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical paper

Out-of-order enabled operating system for uncertain planning, scheduling and execution in aviation maintenance

Fan Yang [a], Cheng Ren [a], Wei Liu [b], Mingxing Li [c], Ming Li [a,d,e,f,g,*]

[a] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
[b] Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong
[c] School of Intelligent Systems Science and Engineering, Jinan University, Zhuhai, China
[d] Research Institute for Advanced Manufacturing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
[e] Collaborative Generative AI (Co-GenAI) Research Centre, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
[f] Research Centre for Digital Transformation of Tourism, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
[g] PolyU Academy for Artificial Intelligence, The Hong Kong Polytechnic University, Hong Kong

A R T I C L E I N F O

Keywords:
Aviation maintenance
Planning
Scheduling and execution (PSE)
Production uncertainty
Operating system
Out-of-order execution (OoOE)

A B S T R A C T

Maintenance has long been a concern in manufacturing, both in the production and product-service phases. As a type of large product, aviation maintenance produces a collection of services to ensure that aircrafts or aircraft systems, components, and structures meet airworthiness standards. Planning, scheduling, and execution (PSE) is important for maintenance systems to optimize resource utilization and job sequencing through decision-making at different time cycles. However, stochastic uncertainty always exists, affecting the stability of the entire maintenance process. Therefore, in this study, which was inspired by operating systems (i.e., Windows, Android, etc.) for processing uncertain user actions with high efficiency, an out-of-order enabled operation system in aviation maintenance (OoO-AMOS) is designed to mitigate the influence of uncertainties that exist in the PSE procedure. Two key components, namely, thread manager and resource manager, are proposed at the kernel level of the OoO-AMOS. The concept of out-of-order (OoO) is deployed for the thread manager to dynamically select the optimal order sequence based on task dependencies and feasibility. A finite state machine (FSM) model is integrated as the operation validation mechanism to formulize the resource states and their transitions. Finally, a case study is conducted to evaluate the effectiveness of the proposed OoO-AMOS. The results show that OoO-AMOS presents significant advantages over traditional approaches. In uncertain environments, the total setup time was reduced by more than 55 %, whereas the maintenance makespan, average order tardiness, and hangar turnover rate achieved improvements of more than 22 %, 31 %, and 23 %, respectively.

Introduction

Maintenance is a critical and necessary part of a manufacturing system, and its goal is to restore a machine component to a condition where it can perform its intended function [1]. In the aviation industry, maintenance produces services of overhaul, repair, inspection, and modification to aircraft systems, components, and structures, which ensure that an aircraft retains an airworthy condition [2,3]. The forecast data released by Oliver Wyman indicate that expansion demand in aviation maintenance will increase 1.8 % annually on average during the next 10 years [4]. Planning, scheduling, and execution (PSE) in aviation maintenance involve decision-making and implementation processes regarding resource utilization and job sequencing optimization [5,6]. Stable operation of the maintenance PSE process is vital for minimizing aircraft downtime, reducing maintenance costs, and increasing aircraft utilization. However, as shown in Table 1, uncertain events exist at each stage of maintenance PSE, which may result in delayed order delivery, low resource utilization, inefficient maintenance, and high downtime costs. Specifically, the planning phase involves developing the maintenance plan and defining workflows and timetables, but temporary order modifications can disrupt the original plan [7]. In the scheduling phase, determining maintenance order priorities and resource allocation is essential but can be challenging due to limited resource availability and potential scheduling conflicts [8]. During the execution phase, operational processing time fluctuations and temporary failures can lead to operational tasks not being

* Corresponding author at: Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong.
E-mail address: ming.li@polyu.edu.hk (M. Li).

https://doi.org/10.1016/j.jmsy.2025.04.011
Received 20 December 2024; Received in revised form 26 March 2025; Accepted 16 April 2025
Available online 25 April 2025
0278-6125/

Table 1
Uncertain events from PSE stages in aviation maintenance.

Phase	Task	Uncertain Event	Possible Consequence
Planning [7]	Tactical plan development	Temporary modification of maintenance orders	Order delivery delays, low resource utilization, low maintenance efficiency, high downtime costs
Scheduling [8]	Task priority ranking and resource allocation	Uncertain resource availability and conflict	
Execution [9]	Complete operational tasks as required	Fluctuating operating duration and temporary breakdown	

successfully completed [9]. Thus, minimizing the influence of uncertainty during the multistage maintenance procedure is essential. However, previous research has focused mainly on how to handle uncertainty during maintenance planning and scheduling processes [10–12]. Therefore, how to reduce the influence of various uncertain events throughout the maintenance process requires further investigation.

In previous research [7,8], historical experiences and patterns were learned from the planning and scheduling stage of aviation maintenance, and forecasting methods were used to minimize the disturbance caused by uncertainties. However, the execution phase is also full of unexpected accidents, such as operator absences and unpredicted processing time [13]. Such uncertain events may not only delay the completion of the task being executed but also disrupt the original maintenance plan and scheduling. Similarly, the process in computer systems is highly uncertain because of the unpredictability of user requirements and the grabbing of limited resources. The operating system (OS) is a fundamental component of a computer system with the ability to respond to and handle uncertain user requirements [14]. The OS senses user requirements and translates them into instructions that the computer can understand and execute. The required resources, such as data, memory, and hardware devices, are then allocated to instructions reasonably according to priorities to ensure the proper functioning of the computer system.

Furthermore, aviation maintenance is a dynamic, complex, multiresource, coordinated process that includes operators, spare parts, hangars, etc. [15]. Advanced technologies equipped with sensing, identification, transmission, and interaction capabilities collect real-time data from PSE stages. These real-time data not only reflect the status of maintenance resources but also capture the inherent uncertainties in the onsite maintenance process. Similarly, the proper functioning of a computer system requires the support of multiple resources. Within the kernel layer of the operating system, thread managers and resource managers serve as the two core functional components. Thread managers are responsible for scheduling thread-level tasks. Out-of-order execution (OoOE) is a dynamic rule for executing instructions in the thread manager according to the real-time availability of resources rather than on the basis of traditional sequential principles [16]. Furthermore, based on real-time status, the resource manager traces the status of the limited resource and rationally allocates according to the instructions' priority to ensure the normal operation of the computer system.

To handle uncertainty during the aviation maintenance multistage, inspired by computer science, this study embarks on the following research questions.

1) Could the concept of an operating system be adopted in an aviation maintenance scenario to mitigate the influence of uncertainty originating from multiple stages?
2) If so, what kinds of underlying mechanisms can thread manager rely on to handle tasks and reduce the influence caused by the uncertainties present in the PSE phases?
3) How can resource manager manage and formulize the availability of multiple resources and perform rational resource allocation with real-time information?

To address the above research questions, this study proposes an out-of-order enabled operation system in aviation maintenance (OoO-AMOS) to facilitate real-time, flexible, and robust PSE decisions in complex and stochastic environments. The main contributions of this research are summarized as follows.

1) Inspired by operating systems (i.e., Windows, Android, etc.), OoO-AMOS is introduced to address uncertainty in aviation maintenance to minimize the impact of uncertain events during the PSE process, which is the first application of the operating system's underlying logic.
2) The OoO-based decision-making mechanism is designed for the thread manager to dynamically select the optimal order sequence on the basis of task dependencies and feasibility, which guarantees that the PSE processes are stable and robust with the occurrence of uncertain events.
3) The resource manager and finite state machine-based operation validation mechanisms are combined to digitally describe the multiresource under uncertainty and visually formulate the dynamic resource status, which enables automatic feedback and real-time state transitions.

The remainder of this article is organized as follows. Section II presents a review of related works. Section III presents the framework of OoO-AMOS. The mathematical model of the OoO principle for thread manager is proposed in Section IV. A case study including system implementation and computational experiments is presented in Section V. Section VI presents a summary of the study with future perspectives.

Literature review

Two research streams are relevant to this research: 1) uncertain PSE in aviation maintenance and 2) OoO-based decision-making methods. These streams of research are reviewed, and the research gaps of this work are summarized.

Uncertain PSE in aviation maintenance

With the development of commercial aviation, an increasing number of scholars are examining ways in which the PSE problem in aviation maintenance can be solved more efficiently. As early as the 1970s, Air Canada developed the aircraft maintenance operations simulation model to replace manual planning and greatly improve the speed of developing maintenance plans [17]. On this basis, numerous mathematical models and system frameworks have been proposed to optimize aviation maintenance planning and scheduling processes [10–12].

Previous studies can be broadly divided into two categories: aviation maintenance planning problems and aviation maintenance scheduling problems. Fleet assignment [18], capacity planning [19,20], and aircraft maintenance routing problems are key topics in aviation maintenance planning. Ben Ahmed et al. [18] proposed a matheuristic consisting of a decomposition approach and a proximity search algorithm to address an integrated airline planning problem. Erkoc et al. [19] constructed an integer programming model for the optimal overhaul process within a limited processing capacity environment. To maximize the total remaining time of a fleet, Başdere et al. developed and solved an integer linear programming model using the branch-and-bound method to determine maintenance routes [21]. For the aviation maintenance scheduling process, maintenance resource scheduling is one of the most

Table 2
Comparison of existing aviation maintenance task solutions.

References	Target	Planning	Scheduling	Execution	Uncertainty
[7]	Maintenance planning in uncertainty	✓			✓
[8]	Technician scheduling		✓		✓
[10]	Long-term maintenance check scheduling	✓	✓		✓
[11]	Maintenance capacity planning and scheduling	✓	✓		✓
[18]	Airline fleet assignment, aircraft routing, crew pairing	✓	✓		✓
[19]	Planning and scheduling for overhaul services	✓	✓		
[20]	Long-term maintenance decision optimization	✓			
[21]	Maintenance route problem	✓			
[22]	Spare part management		✓		
[23]	Aircraft parking layout in hangar		✓		
[24]	Optimizing maintenance schedule and task allocation		✓		
[25]	Long-term maintenance scheduling optimization		✓		✓
[29]	Airline operator management	✓			
[30]	Resource scheduling on multiple on-site task		✓		
[31]	Maintenance manpower planning		✓		
Ours	**Improve aviation maintenance efficiency, minimize the total tardiness and maximize resource utilization**	✓	✓	✓	✓

studied problems. Resources such as maintenance operators, spare parts, and hangars are essential for completing maintenance tasks. Niu et al. [8] addressed the aviation maintenance technician scheduling issue by establishing a model with a practical dynamic task decomposition mechanism, aiming to allocate maintenance workload and ensure timely and effective supply of maintenance personnel. Chen et al. [22] proposed a cyber-physical spare parts intralogistics system from the perspective of spare parts intralogistics. The system's flexible management of aviation spare parts helps to improve resource scheduling. Qin et al. [23] presented a mixed-integer linear programming mathematical model that integrates the interrelations between the maintenance schedule and aircraft parking layout in hangars. In addition, some studies have suggested that there is no clear conceptual separation between maintenance planning and scheduling. For example, Deng et al. [10] developed a practical dynamic programming-based methodology to optimize long-term aircraft maintenance planning. In their subsequent research [24], a decision support system was proposed for task allocation while optimizing aircraft maintenance scheduling. However, current research focuses mainly on the planning and scheduling phases in problem solving and ignores the potential impact of the execution phase on the aviation maintenance process.

Aviation maintenance is a highly complex and dynamic process with varying degrees of uncertainty that significantly impacts the performance of the entire aviation maintenance process [25]. Samaranayake et al. noted that unplanned tasks can constitute up to 50 % of the total maintenance tasks [12]. The occurrence of uncertain events can delay the completion of maintenance tasks, subsequently affecting the start times of subsequent work. A typical system response to uncertainty involves rescheduling the remaining maintenance tasks and operations on the basis of the current situation, which often results in some degree of hysteresis [11]. Therefore, some scholars have tried to utilize forecasting methods to mitigate the influence caused by uncertainties in the aviation maintenance process. Masmoudi et al. [7] categorized uncertainties in the maintenance planning and scheduling phases into tactical and operational levels on the basis of their sources. They established a fuzzy model to address uncertainties and address the impact of uncertainties on maintenance workload at the tactical level due to the uncertainty of macroscopic task contents and at the operational level due to the uncertainty of maintenance task durations. Weide et al. [25] utilized genetic algorithms to generate robust aircraft heavy maintenance schedules in uncertain environments, aiming to reduce the workload and frequency of modifications to maintenance plans. However, unpredictable events also occur in the aviation maintenance execution stage. The above approach, which relies on aviation maintenance historical experience and operational pattern training, is no longer suitable for addressing such problems.

OoO-based decision-making method

OoOE originates from modern computer central processing units, which emphasize the real-time availability of resources and constraints at the time of instruction execution rather than the order of instructions [16]. OoOE provides a method for dynamically analyzing and solving resource-dependent complex logic, avoiding unnecessary pauses in ordered processors and improving the efficiency of instruction processing [26]. Graduation intelligent manufacture system (GiMS) for making decisions under uncertain manufacturing environments has been proposed in recent research [6]. GiMS was also adapted for prefabricated production by Ding et al. [27] as a stable, resilient, and adaptable decision-making mechanism for fixed-point assembly. The underlying logic of GiMS is similar to that of OoOE. They both have the goal of weakening the importance of the original sequence and focusing on the dependencies between operations/instructions and the availability of limited resources. In this context, the principle of OoOE is applied to address uncertainty problems [6,21,23]. Li et al. [9] developed out-of-order synchronization to facilitate flexible, resilient, and coordinated production and intralogistics operations in multiresource-constrained assembly systems. A spatial-temporal out-of-order execution method is composed of real-time planning and scheduling in cyber-physical factories [26]. Sun et al. [28] proposed the out-of-order enabled dueling deep Q network approach for dynamic additive manufacturing scheduling.

Research gap summary

According to the literature reviewed above, Table 2 provides an overview of research investigations into the aviation maintenance PSE uncertainty problem. Although existing research has made significant contributions in the field of aviation maintenance uncertainty and out-of-order-based decision-making methods, there are still some research gaps that need to be bridged.

1) Forecasting methods, which employ historical experiences and patterns of maintenance planning and scheduling processes, were used to minimize the impact caused by uncertainties. However, current research lacks an approach for responding to predictable and unpredictable uncertainties from the whole process of aviation maintenance PSE.
2) Currently, out-of-order (OoO)-inspired approaches have been applied in production scenarios and have demonstrated excellent performance in addressing uncertainties in the PSE process. However, although aviation maintenance scenarios are also characterized as multistage, multiresource, and dynamic, existing research has

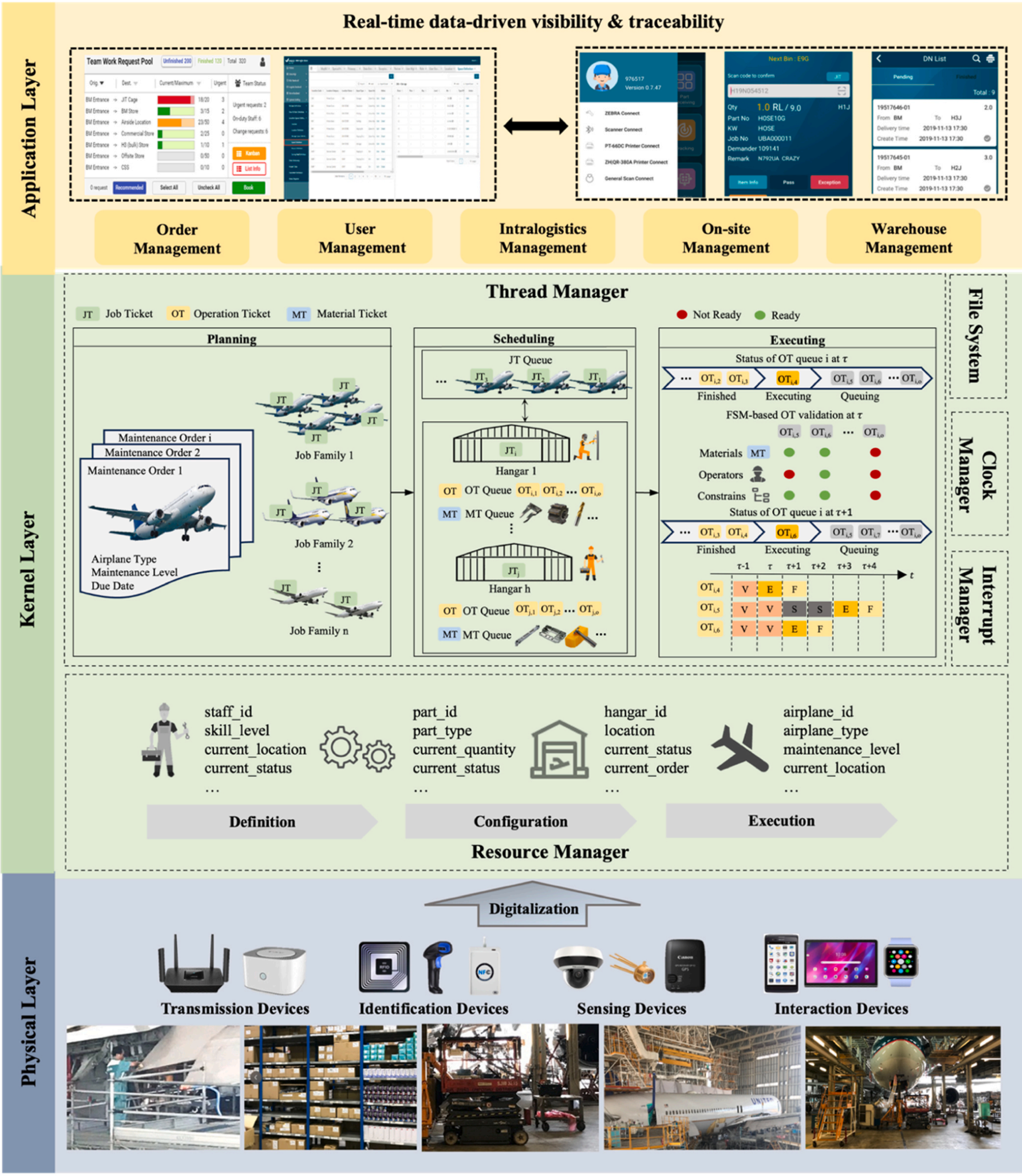

Fig. 1. Overall OoO-AMOS framework.

seldom applied OoO principles to address uncertain PSE in aviation maintenance.

To bridge these gaps, in this research, an OoO-based aviation maintenance operating system (OoO-AMOS) is used to address the uncertainties during the PSE phases. Specifically, the uncertainties in the PSE stages can be detected by IoT devices with sensing, identification, transmission, and interaction functions. After detection, an OoO-based decision-making mechanism is designed and adopted to handle the uncertainties in each stage.

Overall framework of OoO-AMOS

An operating system is a fundamental component of computer systems and is positioned between the underlying hardware and users. It manages hardware resources at the lower level and provides services to users at the higher level [14]. The process in a computer is highly uncertain due to the unpredictability of the user's operations. In the application layer, users initiate requests through applications. Once the operating system detects a user request, it requests the component in the kernel layer to execute tasks by system calls. In the kernel layer, the thread manager processes instructions based on various scheduling algorithms, such as out-of-order execution, first-come-first-service, and shortest-job-first. OoOE is a real-time execution method that takes into account the availability of data, resources, and constraints [16]. By employing OoOE rules in the thread manager, the operating system dynamically resolves blocking issues caused by uncertainties. The resource manager is responsible for managing and coordinating all the resources needed for the operation of the operating system, such as data, hardware devices, memory, etc. It has the functions of tracking the status of resources and optimizing the use of resources. Based on the real-time status, the resource manager rationally allocates limited resources to instructions according to their priority to ensure the normal operation of the computer system. Finally, device drivers interact with hardware to fulfill user requirements.

Inspired by the operating system in computer science, this section provides a comprehensive introduction to how OoO-AMOS is deployed in aviation maintenance scenarios to handle uncertainties. Fig. 1 illustrates the proposed framework of OoO-AMOS. The framework is divided into three main layers: physical layer, kernel layer, and application layer.

Physical layer in OoO-AMOS

The first component involves the management of physical resources within the OoO-AMOS. Advanced technologies equipped with sensing, identification, transmission, and interaction capabilities are deployed in practical maintenance scenarios to collect real-time data, facilitating communication between physical and digital objects [27]. Various types of sensing devices are deployed in different locations within hangars to perceive signals and events in the maintenance environment, which are then converted into digital signals to provide data and information for real-time decision-making [22]. Identification labels such as RFID, QR codes, NFC, etc., are used to bind and create intelligent entities for resources such as humans, spare parts, and equipment. By reading the identification and verification of devices, real-time tracking of the resource status is achieved. The physical objects within aviation maintenance are interconnected and exchange collected real-time data through network gateways, routers, and other devices with transmission capabilities. Finally, data communication is facilitated through interactive devices such as mobile and wearable devices. The digitalization process creates corresponding digital objects for each physical object, abstracting physical objects into digitized information to capture the uncertainties present in aviation maintenance [32,33].

Kernel layer in OoO-AMOS

The second component is the kernel layer, which serves as the core of the entire framework. It consists of the resource manager, thread manager, and other supporting kernel services.

Resource manager

PSE in aviation maintenance is a multiresource coordinated process that involves operators, spare parts, hangars, and other maintenance resources. The real-time based maintenance resource manager is responsible for managing digital resources, describing the resource status, and analyzing potential uncertainties. First, the resource manager utilizes the real-time data collected from IoT devices to construct digital representations of objects using a series of relevant information. This includes the object ID, attributes (e.g., level, type, location), and real-time status (e.g., current location, order status, completion progress, part quantity). Second, the resource manager facilitates comprehensive visibility and traceability of multiple resource status by providing real-time monitoring, detailed state tracking, and performance assessments. This ensures efficient resource utilization, timely identification of issues, and informed decision-making across the aviation maintenance PSE process. Third, by analyzing real-time data, it becomes possible to identify and address some of the uncertainties hidden in the maintenance process. This proactive approach not only helps in detecting potential issues at an early stage but also enables timely interventions, reducing the risk of disruptions and ensuring the overall efficiency and reliability of the process.

Thread manager

The thread manager, which is based on the principles of OoO, is responsible for managing and coordinating the processing of tasks in the aviation maintenance PSE procedure. To organize the thread manager, three types of functional tickets are introduced: job tickets (JTs), material tickets (MTs), and operation tickets (OTs). JT is associated with each maintenance task within the customer orders. Each JT is linked to a series of MTs and OTs. MTs are generated based on the bill of materials (BOM) and include tools, spare parts, equipment, and other required maintenance materials. OTs represent specific operational tasks derived from the decomposition of each JT. During the planning phase, customer orders randomly arrive and enter the order pool, where each maintenance task within the order is assigned a JT. Each JT contains information such as airplane type, maintenance task description, and due date. Similar JTs are grouped into a job family (JF) based on different attributes. In the scheduling phase, JFs are assigned to the most suitable and available hangars based on real-time status. JTs within each JF enter the matched hangars for maintenance tasks according to their priorities. Prior to entering the hangar, each hangar needs to configure its internal settings based on aircraft type and task content. Once a JT enters the hangar, it proceeds to the execution phase to commence the maintenance task. The FSM-based OT validation mechanism is employed to assess three conditions based on real-time data: (1) Are the required MTs available? (2) Have the required operators arrived? (3) Are the task sequencing restrictions satisfied?

Supporting kernel services

There are also other functional components, such as the clock manager, file system, and interrupt manager, are necessary for supporting the proper functioning of the kernel layer.

1) The clock manager is involved in flexibly decomposing the total maintenance time horizon H into three granularities based on different PSE phases. The long-term time horizon, denoted as T, represents the total time required to complete all tasks within a JF. It provides an overall view of the time needed for the entire JF. The midterm time horizon, denoted as t, indicates the duration required to complete a single JT. It focuses on the time needed to accomplish the tasks associated with a specific JT. The short-term time horizon, denoted as τ, represents the processing time for each OT. It reflects the time required to complete individual operational tasks. Importantly, the lengths of these three types of time horizons are not strictly defined but depend on the specific processing times associated with the tasks.
2) The file system primarily manages the tickets generated during the aviation maintenance process. For example, when an order is received, the corresponding BOM information is stored in the file system. When the intralogistics department receives a shipment notification, it retrieves the relevant BOM from the file system to arrange the intralogistics task. The file system handles the storage, retrieval, and management of various documents and information related to the maintenance process.
3) The interrupt manager responds to and initiates interrupt requests when the system encounters anomalies, leading to the suspension of

Table 3
Notation and descriptions.

Notations	Descriptions
JT_i	Job ticket of maintenance job i
$OT_{i,o}$	Operation ticket of operation o of job JT_i
$MT_{i,m}$	Material ticket of material m required by job JT_i
$MT^{wh}_{i,m}$	Warehouse of required material $MT_{i,m}$
JTP^T	JT pool in T
JTQ^t	JT queue in t $JTQ^t = JTQ^t_{finished} \cup JTQ^t_{executing} \cup JTQ^t_{queuing}$
OTQ^τ	OT queue at τ $OTQ^\tau = OTQ^\tau_{finished} \cup OTQ^\tau_{executing} \cup OTQ^\tau_{queuing}$
MTQ^τ	MT queue at τ $MTQ^\tau = MTQ^\tau_{finished} \cup MTQ^\tau_{executing} \cup MTQ^\tau_{queuing}$
dt_i	Due date of JT_i
JF_c	Job family c
n_c	Number of jobs in JF_c
spt_i	Standard processing time of JT_i
at_i	Arrive time of JT_i
ct_i	Completion time of JT_i
JF^T_c	Job family c in T
FD^T_c	Family due date of JF^T_c
rpt^τ_i	Remaining processing time of JT_i at τ
$rpt^\tau_{i,o}$	Remaining processing time of $OT_{i,o}$ at τ
$rt^{T,t}_h$	Estimated ready time of hangar h in time window (T,t)
imp_i	The importance of JT_i
$n^{m,t}_i$	Number of ready materials of JT_i in time window (T,t)
$n^{m,\tau}_i$	Number of ready materials of jt_i at τ
n^T_i	Number of JTs that belong to the same JF_c with JT_i
$st^h_{i-1,i}$	Setup time of hangar h between JT_{i-1} and JT_i which are not in same JF_c
f^h	Setup time of hangar h between jt in same JF
$n^\tau_{avapo,o}$	Number of available operators for $OT_{i,o}$ at τ

ongoing tasks within the hangar. It then performs appropriate checks and adjusts actions in response to the interrupt cause.

Application layer in OoO-AMOS

The third component is the application layer, which supports user-friendly interactions for all stakeholders involved in aviation maintenance, such as operators, managers, and customers. For example, operators on the shop floor utilize mobile devices to access task lists and upload their work status, and managers access real-time and historical data through visual tools on their computer terminals to guide and optimize their decision-making processes. The application layer encompasses several key components, including the following:

- Order Management: Complete functions such as creating, processing, and tracking orders, as well as providing statistics, reminders, and notifications.
- User Management: Manage user accounts, permissions, and roles and provide user login, registration, and authentication.
- Intralogistics management: Track and manage the movement of items within the maintenance environment, including inbound, outbound, and transfer operations.
- Site Management: Coordinate and manage site work, including task assignment, progress tracking, resource scheduling, and problem solving.
- Warehouse Management: Manage warehouse inventory, including functions such as inventory tracking, goods receiving and shipping, inventory adjustment, and reporting.

The orders released by users at the application layer are uncertain. OoO-AMOS receives these user orders and generates maintenance requests at the kernel layer. The thread manager converts all tasks associated with an order into JTs, and the PSE operations are processed based on the OoO principle. During the maintenance process, the resource manager drives supporting departments to supply the necessary maintenance resources to the JTs based on real-time information.

OoO-enabled thread manager in AMOS

In the previous section, the workflow of the thread manager in OoO-AMOS was described. In this section, a detailed explanation of the decision rules, which are employed in each stage of the PSE in the thread manager, is provided. All the notations and descriptions are presented in Table 3.

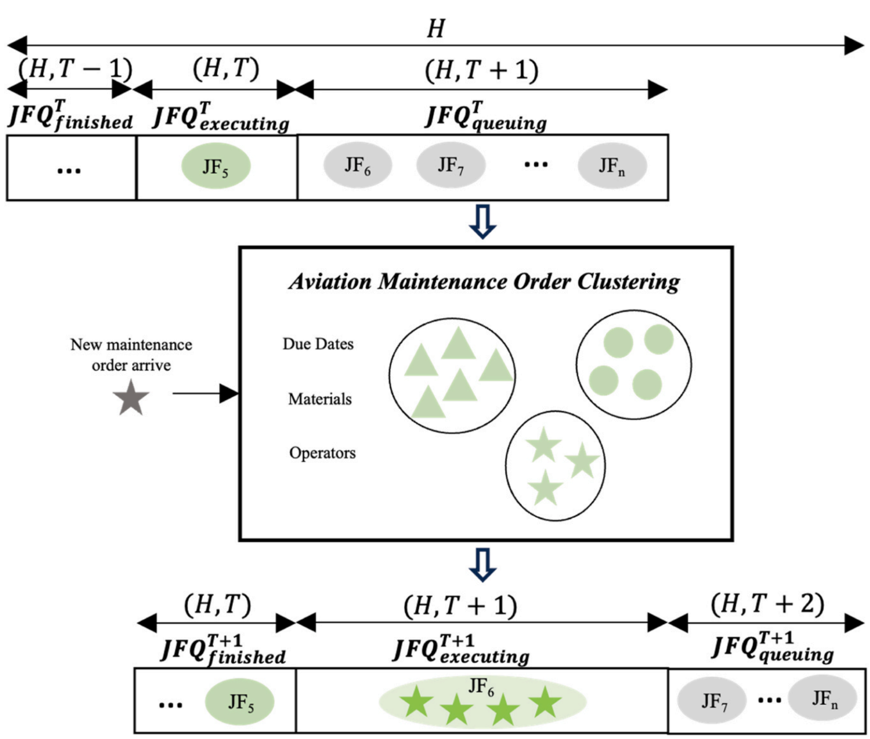

Fig. 2. Spectral clustering for the maintenance planning phase.

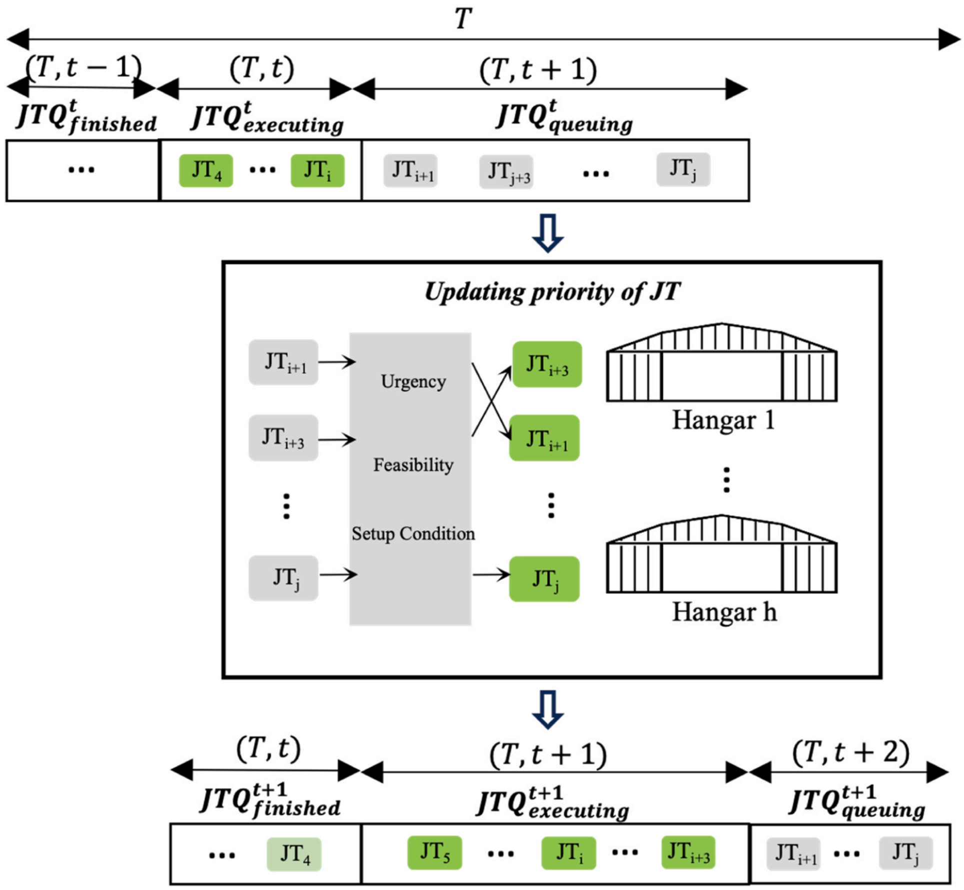

Fig. 3. Updating priority of JT in the maintenance scheduling phase.

Spectral clustering for the planning phase

Inspired by GiMS [6,34,35], in the aviation maintenance planning phase, spectral clustering is employed to calculate the similarity and cluster the orders/JTs within the JT pool at T (JTP^T). After the clustering process, all the JFs are queued for scheduling and execution. The JF queue at T is defined as

$$JFQ^T = JFQ^T_{finished} \cup JFQ^T_{executing} \cup JFQ^T_{queuing} \quad .\# \tag{1}$$

Through the utilization of this approach, similar orders/JTs are grouped together in the same cluster JF and released into the hangar within the same time horizon T. The purpose of clustering similar orders/JTs is to minimize setup times within a JF without prioritizing the specific processing sequence, thereby enabling out-of-order planning. Consequently, it provides flexibility to address any uncertainties that may arise during the planning phase. For the example in Fig. 2, a new maintenance order arrives at time (H, T). According to the clustering results of JFQ^T at that time, the new order is assigned to the most suitable JF_6 based on its attributes. At time $(H, T+1)$, JF_6 is executed sequentially, including the new order.

In this research, two attributes of the orders are used to calculate the distance for clustering: the due date and the required materials.

The due date refers to the delivery date of each JT. Measuring the distance between due dates helps meet maintenance delivery deadlines and enhances customer satisfaction. The due date distance $dist^d_{ij}$ between JT_i and JT_j is measured by calculating the standardized Euclidean distance [36]

$$dist^d_{ij} = \sqrt{\left(\frac{d_i - d_j}{std(d)}\right)^2} .\# \tag{2}$$

The required materials refer to the bill of materials (BOM) associated with each JT, including the spare parts, tools, and machines required for each maintenance task. In aviation maintenance, the required materials exhibit a high degree of specialization, wherein different types of aircraft often necessitate distinct materials, where MT^{wh}_i represents the set of warehouses for storing the required material of JT_i. Evaluating the similarity between warehouses that stock the required materials enhances intralogistics efficiency and reduces the setup time between JTs in the hangar. The required material distance $dist^m_{ij}$ between JT_i and JT_j is calculated by the Jaccard similarity coefficient [37]

$$dist^m_{ij} = 1 - \frac{\left|MT^{wh}_i \cap MT^{wh}_j\right|}{\left|MT^{wh}_i \cup MT^{wh}_j\right|} .\# \tag{3}$$

The overall distance $Dist^T_{ij}$ between JT_i and JT_j is

$$Dist^T_{ij} = \omega^d_{dist} \bullet norm(dist^d_{ij}) + \omega^m_{dist} \bullet norm(dist^m_{ij}) .\# \tag{4}$$

where ω^d_{dist} and ω^m_{dist} are the weights of the due date and the required material distances, respectively. Notably, the distance calculation is highly flexible. The weights can be customized according to actual requirements and situations.

By calculating the distance between each pair of JTs in the order pool in T, a $\left|JT^T\right| \times \left|JT^T\right|$ symmetric distance matrix is obtained. This distance matrix is used in the subsequent spectral clustering algorithm. Spectral clustering is a graph-based clustering algorithm. In the aviation maintenance context, each JT is represented as a node in an undirected graph, and each edge represents a pair of JTs, with the edge weight being the distance between each pair of JTs. The spectral clustering algorithm, which is based on the work of Von Luxburg [38], classifies JTs into

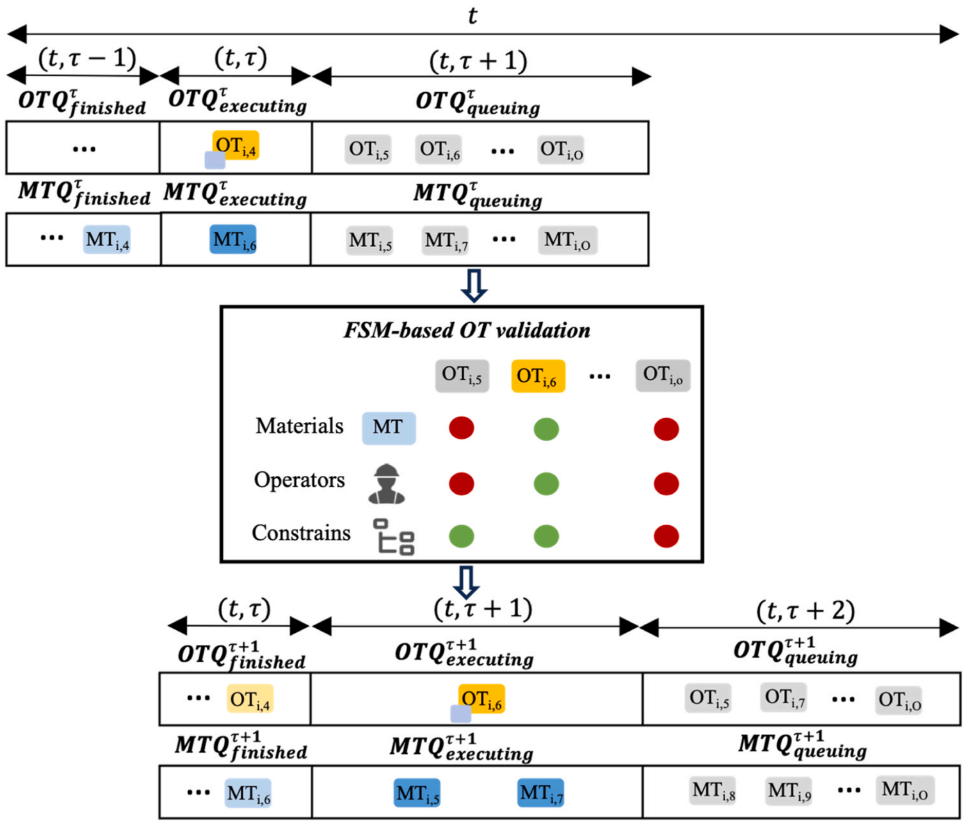

Fig. 4. FSM-based validation mechanism for the execution phase.

different JFs.

Suitability-based priority updating for the scheduling phase

During the planning phase, JTs are clustered into multiple JFs based on the similarity of their order attributes. In the scheduling phase, it is necessary to allocate appropriate hangars for each JF and determine the priority of the JTs within them. Furthermore, the scheduling process is dynamic and responsive to real-time data. As real-time information becomes available, such as setup time changes or resources arriving at the onsite buffer, the scheduling decisions can be adjusted accordingly. This feedback loop ensures that the scheduling process remains adaptive and optimized based on the evolving conditions within the maintenance process.

Considering the real-time suitability of both JFs and hangars to achieve appropriate matching between them is essential. The family due date (FD) of JTs within a JF reflects the overall urgency of the maintenance orders associated with each JF. The FD^T_c of JF^T_c is defined as

$$FD^T_c = \frac{\sum_{JT_i \in JF^T_c} d_i}{n^T_c}.\# \tag{5}$$

JFs with earlier FDs are considered higher priorities, as their completion is crucial to meeting deadlines and ensuring customer satisfaction. High-priority JFs are matched to the earliest available hangars to minimize the hangar idle time. For hangars, their available time is determined by the completion time of the last JT in the matched JF. It's important to note that the available time of a hangar is not a fixed value but is adjusted based on real-time data after the end of the time horizon in different dimensions.

Once the matching between JFs and hangars is completed, real-time prioritization of JTs within a JF is required to identify the most suitable JT to start executing within the time horizon (T,t). Fig. 3 illustrates the process of updating the JT priorities from (T,t) to $(T,t+1)$. At time (T, t), JT_{i+1} has a higher priority than JT_{i+3}, which both are waiting to be executed in the $JTQ^t_{queuing}$. However, at time $(T,t+1)$, due to the impact of uncertainty at the scheduling layer, the priority of JT_{i+3} increases. As a result, at time $(T,t+1)$, JT_{i+3} is executed first, whereas JT_{i+1} continues to wait in the queue.

To comprehensively evaluate the real-time priority of JTs, this study considers the following three aspects:

Urgency of JT: The due date d_i, standard processing time spt_i and estimated ready time of hangar h $rt^{T,t}_h$ of JT_i and the importance of the associated customer imp_i are used to describe the urgency of JT_i. The higher the urgency is, the higher the priority of the JT. The priority of JT_i in terms of urgency $prio^{ur,t}_i$ in time horizon (T,t) is calculated as

$$prio^{ur,t}_i = \left(d_i - spt_i - rt^{T,t}_h\right) * imp_i.\# \tag{6}$$

On-site feasibility: Since MTs need to be executed and completed before related OTs, the completion of MTs indicates that the required materials for the OTs are available. Therefore, a higher number of completed MTs suggests that the JT is more executable and can be prioritized accordingly. The number of ready materials of size JT_i in time window(T,t) $n^{m,t}_i$ is set to finish $MT_{i,m}$ in time window(t,τ), which can be calculated as follows:

$$n^{m,t}_i = \sum_{\tau \subset t} n^{m,\tau}_i.\# \tag{7}$$

The priority of jt_i in terms of onsite feasibility $prio^{os,t}_i$ in time horizon (T,t) is calculated as

$$prio^{os,t}_i = 1 - \frac{n^{m,t}_i}{|MT_i|}.\# \tag{8}$$

Complexity of setup conditions: The state of the hangar at time (T, t-1) is considered. The shorter the setup time $st^{h,t-1}_i$ between the JT and the orders executed at (T, t-1), the higher the priority of the JT. The priority

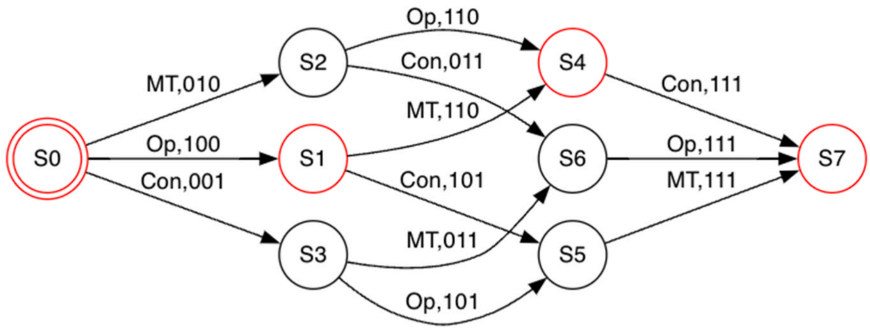

Fig. 5. State transition of FSM-based OT validation.

of jt_i in terms of setup condition $prio_i^{se,t}$ in time horizon (T, t) is calculated as

$$prio_i^{se,t} = \begin{cases} 0, \textit{if setup work is not required} \\ st_i^{h,t-1}, \textit{if setup work is required} \end{cases}.\# \tag{9}$$

The setup time is contingent upon the conditions of the two preceding and succeeding *JTs*. In cases where the two *JTs* belong to the same *JF*, the hangar requires only basic organizational tasks that need f^h to complete the setup job. Conversely, if the two *JTs* do not belong to the same *JF*, an extensive setup of the hangar is essential, typically necessitating a long setup time. The formula utilized to describe the setup time is outlined below:

$$st_{i-1,i}^h = \begin{cases} f^h, \textit{if} jt_{i-1} \textit{and} jt_i \textit{are in the same JF} \\ st_{i-1,i}^h, \textit{if} jt_{i-1} \textit{and} jt_i \textit{are not in the same JF} \end{cases}.\# \tag{10}$$

The overall priority of jt_i is $Prio_i^t$, which can be calculated by

$$Prio_i^t = \omega_{prio}^{ur} \bullet norm(prio_i^{ur,t}) + \omega_{prio}^{os} \bullet norm(prio_i^{os,t}) + \omega_{prio}^{se} \bullet norm(prio_i^{se,t}) \# \tag{11}$$

where ω_{prio}^{ur}, ω_{prio}^{os} and ω_{prio}^{se} are the weights of different priorities. It is important to highlight that the priority calculation offers significant flexibility. Thus, the weights and indexes can be tailored to meet specific needs and adapt to varying conditions.

FSM-based OT validation mechanism for the execution phase

The execution phase of aviation maintenance is a process that operates at a fine-grained level and needs to make decisions from the perspective of OTs. Inspired by the concept of OoOE, only the OTs that satisfy both the availability of resources and the constraints are executed [26]. Maintenance execution primarily encompasses two main tasks. Firstly, based on real-time data within the maintenance process, resources such as operators, spare parts, tools, etc., are virtually allocated to different OTs. Subsequently, an OT validation mechanism based on a finite state machine is employed to assess the validity of OTs at different time intervals.

Fig. 4 shows the process of FSM-based OT validation in (t, τ) and the result of the OT queue changing in $(t, \tau + 1)$. At time (t, τ), $OT_{i,4}$ is currently being executed. At the same time, $OT_{i,5}$ and $OT_{i,6}$ are waiting in the OT queue. At this point, all OTs in the OT queue need to undergo an FSM-based validation mechanism to check if the conditions for execution are satisfied. However, $OT_{i,5}$ cannot be executed because the required operator and materials have not yet arrived at the maintenance site, whereas $OT_{i,6}$ passed the validation with all the required resources and constraints. Therefore, at time $(t, \tau+1)$, $OT_{i,6}$ with higher priority is executed first, and $OT_{i,5}$ remains in the OT queue waiting for the next validation.

Mealy machine, as one type of FSM, can be adopted to enhance real-time data utilization and formulate the OT validation process [39]. The FSM-based OT validation mechanism is defined as a six-tuple $(S, s_0, \Sigma, \Lambda, T, G)$.

where $S = \{s_0, s_1, s_2, \ldots, s_n\}$ represents the set of finite states, which represents the state of the conditions affecting the validity of OT.

where s_0 is the initial state, which means that the corresponding operation does not have any condition to be fulfilled.

$\Sigma = \{\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_n\}$ is the set of input alphabets that represent the set of input elements required for operation and the preceding operational constraints.

$\Lambda = \{\lambda_0, \lambda_1, \ldots, \lambda_n\}$ is the set of output alphabets, and $G : S \times \Sigma \to \Lambda$ is the set of output functions.

$T : S \times \Sigma \to S$ is a set of transition functions. In the FSM, the state transition is determined by the transition function, which is related to the present state and the input signal.

$\delta(p, l) = q$ where $p, q \in S$ denotes that in state p, with input signal l, the state migrates from p to q.

To represent the validation state of $OT_{p,j}$ in terms of an FSM, there is no sequence among the three conditions. $OT_{p,j}$ can be recognized as valid or ready to be executed only if all three conditions are satisfied. Thus, an 8-state FSM is designed, where each state represents a possible combination of condition states. A state can be represented as a four-bit binary number, where each bit represents whether a condition is satisfied or not. When a condition is satisfied, the value of the corresponding bit changes from 0 to 1. Each round of state transitions is described in detail in Fig. 5. The transition function depends on the current state and input. For example, if the current state is s_0 and the input is the operators, the next state would be state s_1. The transition function can be $\delta(s_0, \sigma_0, s_1)$. Under these circumstances, if another input is the MT, it jumps to state s_4. The transition function can be $\delta(s_2, \sigma_1, s_5)$. The operation is not valid until the state is transferred to s_7.

Each condition is validated as follows:

Availability of material. Verify if the required materials for $OT_{i,o}$ have been delivered to the responding hangar and finished in $MTQ_{finished}^{\tau}$ atτ

$$MT_{i,m} \in MTQ_{finished}^{\tau}.\# \tag{12}$$

Availability of operator. Verify the presence of at least one operator capable of conducting the maintenance operation $OT_{i,o}$ at timeτ

$$n_{avapo,o}^{\tau} \geq 1 \tag{13}$$

where $n_{avapo,o}^{\tau}$ is the number of available operators for $OT_{i,o}$ at τ.

Satisfaction of the precedence constraints. Verify the completion of the OTs that are required to precede $OT_{i,o}$ at time τ. A constraint matrix $PC_{o,o'}^{\tau}$ is introduced to represent the operational precedence constraints

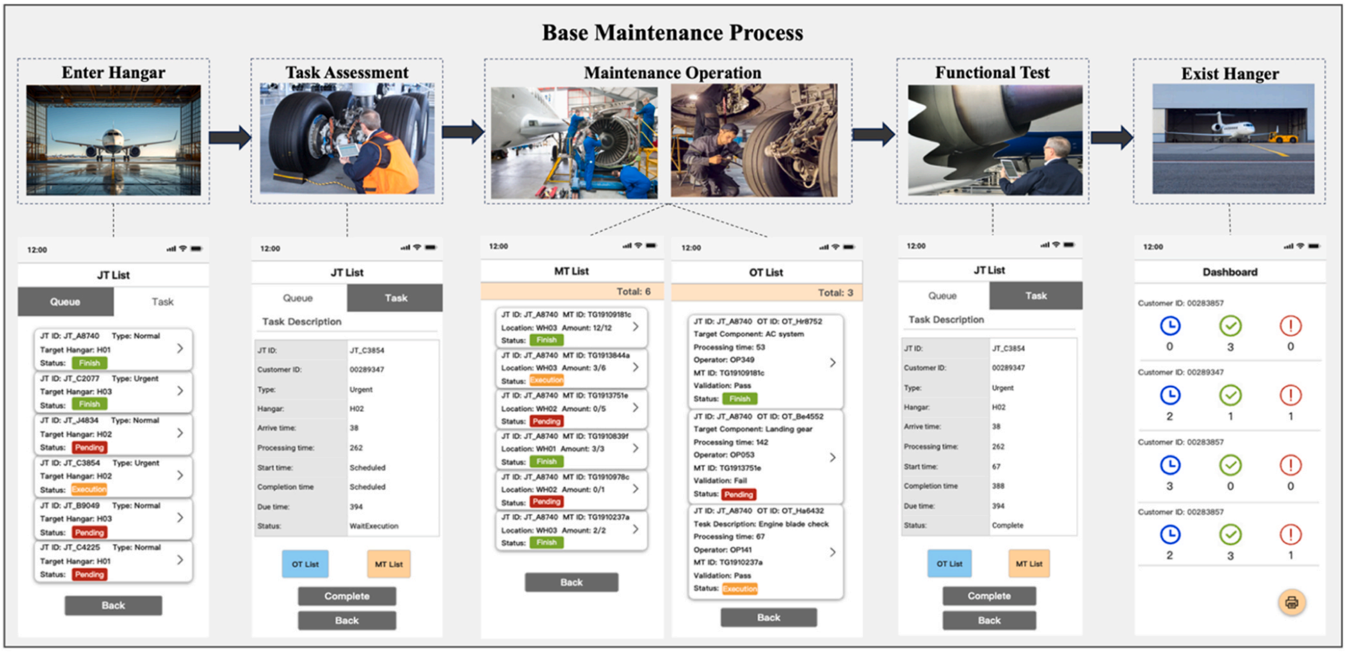

Fig. 6. OoO-enabled base maintenance process.

$$PC^{\tau}_{o,o'} = \begin{pmatrix} 0 & C^{\tau}_{1,2} & \cdots & C^{\tau}_{1,O} \\ C^{\tau}_{2,1} & 0 & & C^{\tau}_{2,O} \\ \vdots & & \ddots & \vdots \\ C^{\tau}_{O,1} & C^{\tau}_{O,2} & \cdots & 0 \end{pmatrix}, \# \tag{14}$$

$$C^{\tau}_{o,o'} = \begin{cases} \&0, & OT_{i,o} \text{does not necessarily precede} OT_{i,o'} \\ \&1, & OT_{i,o} \text{must precede} OT_{i,o'} \end{cases} .\# \tag{15}$$

Case study

A case study from a collaborative aircraft engineering company based in Hong Kong was introduced in this section to validate the effectiveness of the proposed OoO-AMOS. There are two parts: 1) implementation of OoO-AMOS in the aviation maintenance scenario and 2) numerical experiments.

Implementation of OoO-AMOS in an aviation maintenance scenario

Motivated by the team's research projects in the aviation maintenance field, the OoO-AMOS has been implemented in the maintenance department of a collaborative company. The maintenance process involves dismantling aircraft components, such as the fuselage, engines, and landing gear, for repair or replacement. Thus, the process must be conducted entirely within the hangar, and no movements outside the hangar are permitted during maintenance processes. Due to the limitations with the partner company, this study was only able to complete the validation experiments in a single hangar scenario.

Following the formulation in Section 3 and Section 4, an OoO-AMOS enabled application was developed via the Java programming language in the Java runtime environment 8.0. Tomcat 7.0 was used as the web application server, and SQL Server 2012 was used as the database server. A mobile application was used with Android 11.0. Fig. 6 illustrates the full maintenance process in the hangar. Upon an aircraft entering the hangar, inspection personnel undertake a task assessment according to the requirements outlined in the customer orders. Then, the intralogistics and maintenance departments receive execution requests for a set of MTs and OTs. During the execution operations, real-time onsite data are collected, analyzed, and presented. Customers, maintenance managers, and operators monitor the entire maintenance cycle via smart and wearable devices. Once all maintenance operations are completed, maintenance managers conduct functional testing of the aircraft. After successfully passing the functional tests, the aircraft is released from the hangar and delivered to the customers.

Numerical experiments

A numerical experiment was conducted to further validate the effectiveness of the OoO-enabled decision-making mechanism on a theoretical level. For ease of representation, OoO will be used in place of the OoO-enabled decision-making mechanism.

Experiment settings

The historical data used in this study was obtained from the collaborative company. As the Table 4 shows, through data analysis, the dataset can be categorized into three main types: Job Data, Operation Data, and Material Data, which correspond to the core information of maintenance tasks, operational processes, and material management, respectively.

Based on previous similar studies and real historical data, the experimental datasets were generated according to Table 5 [6,40,41]. Six schemes, organized in a 2× 3 matrix, were designed to simulate the performance of OoO under certain and uncertain environments with varying order volumes. Additionally, Table 6 shows that four different types of uncertain events existing in PSE were introduced in the experiments, including urgent customer orders, stochastic setup/processing time, stochastic material delivery time, and equipment failure [42]. To ensure a reliable level of experimental results, all performance evaluation experiments based on the logic of Algorithm 1 were repeated 100 times. All the experiments were conducted using custom code in PyCharm.

Algorithm 1. OoO-based MRO-PSE decision making mechanism

Input: current time τ, JTP^T
Output: JT completion queue $JTQ_{\text{completion}}$

Grouping the $JT_i \in JTP^T$ into different JF by spectral clustering algorithm, $JFP = \{JF_1, JF_2, \ldots, JF_c, \ldots, JF_f\}$;
for $JF_c \in JFP$ **do**
  Calculate the fdt_c of JF_c by Eq(4);
  for $JT_i \in JF_c$ **do**
    Calculate the Prio_i^t of JT_i by Eq(5-8)
  end
end
Initialize

$$JF_{\text{ing}} \leftarrow \underset{JF \in JFP}{\text{argmin}}\, fdt(JF),$$
$$JT_{\text{ing}} \leftarrow \underset{JT \in JF_{\text{ing}}}{\text{argmin}}\, \text{Prio}_i^t(JT),$$
$$OT_{\text{ing}} \leftarrow \forall OT \in OTQ_{\text{valid}}$$

if $JF_{ing} == \phi$ **then**
  $JFP \setminus JF_{\text{ing}}$;
  Start the hangar setup process;
  Check for urgent order arriving and update fdt of each JF;
  Set the new JF_{ing}, $JF_{\text{ing}} \leftarrow \underset{JF \in JFP}{\text{argmin}}\, fdt(JF)$;
  $\tau \leftarrow \tau + 1$;
  Continue;
end
else
  if $JT_{ing} == \phi$ **then**
    $ct_{\text{ing}} \leftarrow \tau$;
    $JF_{\text{ing}} \setminus JT_{\text{ing}}$ and $JT_{\text{ing}} \bigcup JTQ_{\text{completion}}$;
    Update the Prio_i^t of each JT;
    Set the new JT_{ing}, $JT_{\text{ing}} \leftarrow \underset{JT \in JF_{\text{ing}}}{\text{argmin}}\, \text{Prio}_i^t(JT)$;
    $\tau \leftarrow \tau + 1$;
    Continue;
  end
  else
    if OT_{ing} *is not finished* **then**
      $\tau \leftarrow \tau + 1$;
      Continue;
    end
    else
      Search valid OT by FSM-based OT validation by Eq(13-16);
      if $OTQ_{valid}^{\tau} == \phi$ **then**
        $\tau \leftarrow \tau + 1$;
        Continue;
      end
      else
        Set the new OT_{ing}, $OT_{\text{ing}} \leftarrow \forall OT \in OTQ_{\text{valid}}^{\tau}$
      end
    end
  end
end
return $JTQ_{\text{completion}}$

Table 4
Structure of the base maintenance process historical data.

Data Type	Description	Sample
Job Data		
Job ID	Unique identifier for each maintenance job.	JT_C3854
Customer ID	Identifier for the customer.	00289449
Airplane Type	Type or model of the airplane being serviced.	BOE737
Priority	Priority level of the job (e.g., high, medium, low).	High
Processing Time	Estimated time required to complete the job.	120
Delivery Date	Scheduled date for job completion and delivery.	485
Arriving Time	Time when the airplane arrives at the maintenance facility.	283
Leaving Time	Time when the airplane departs from the maintenance facility.	356
BOM	Bill of Materials.	[TG1910237c, TG2843558a,…]
Status	Current status of the job (e.g., pending, in progress, completed).	Completed
Operation Data		
Operation ID	Unique identifier for each operation within a job.	OT_Be4552
Operation Component	Description of operation performed (e.g., inspection, repair, replacement).	Engine
Operator	Identifier for the technician or team performing the operation.	OP053
Part ID	Unique identifier for each part or material performing the operation.	TG2843558a
Starting Time	Time when the operation begins.	313
Finishing Time	Time when the operation is completed.	325
Status	Current status of the operation (e.g., not started, in progress, completed).	Completed
Material Data		
Part ID	Unique identifier for each part or material.	TG2843558a
Location ID	Identifier for the storage location of the part.	WH-A12
Time OUT	Time when the part is checked out from the inventory.	298
Time IN	Time when the part is returned to the inventory (if applicable).	309
Status	Current status of the part (e.g., in use and available).	Available

Table 5
Experimental dataset.

Data	Values
Number of customer order	30, 50, 100
Number of jobs per customer order	$U[1,3]$
Number of job family	$\lceil \lvert JT\rvert/3 \rceil$
Number of part types in BOM	10
Number of variants of each part type	$U[1,10]$
Standard processing time of each part of variants	$(5,4,2,4,4,1,1,5,4,5)/h$
Number of operations	For small-scale jt:$U[1,3]$ For mid-scale jt:$U[4,6]$ For mid-scale jt:$U[7,10]$
Operational constraints	Operation 1 must be the first. No operation constraints between operation 2–9. Operation 10 must be the last.
Hangar setup time	For jt not in same JF:$U[30,50]/h$For jt in same JF:$f^h = 5/h$
Mean time between arrivals	$MBTA = 100/h$
Order arrive timeat_i	$Exp(MTBA)$
Order due timedt_i	$at_i + \alpha * \sum_{i\in co} spt_i, \alpha \in [3.0, 5.0]$

Table 6
Uncertain events generation.

Data	Values
Urgent customer orders	
Number of urgent orders	$\frac{1}{2}\times \lvert JT\rvert$
Due time of urgent orders	$at_i + \alpha * \sum_{i\in co} spt_i, \alpha \in [1.5, 3.0]$
Stochastic setup/processing time	
Variation coefficient	$cv = 0.3$
Standard	p
Actual job setup/processing time	$U[(1-cv)p,\ (1+cv)p]$
Stochastics material arrive time	
Mean time between material arrive	MTBM = 50/h
Material arrive time	Exp (MBTM)
Equipment Failure	
Mean time between failure	$MTBF = 1000/h$
Failure happen time	$Exp(MTBF)$
Time for repair	$U[30,50]/h$

Performance measurement and benchmark

As presented in Table 7, four typical and common rules Earliest Due Date (EDD), Shortest Processing Time (SPT), Shortest Setup Time (SST), and Just-in-Time (JIT) were adopted in the experiments to evaluate the performance of the OoO for decision-making in PSE [43]. Moreover, four measures have been used to evaluate performance [44].

- Makespan (MS): This refers to the total amount of time needed to completely process all *JTs*. A minimum makespan usually indicates high maintenance efficiency

$$MS = \max_j \{ct_j\}.\# \tag{14}$$

- Total setup time (TST): This is the total setup time between each *JTs*. A lower TST implies that fewer nonvalue-added operations occur during the maintenance procedure.

$$TST = \sum_{i=2}^{|JT|} st_{i-1,i}^h.\# \tag{15}$$

- Mean order tardiness (MOT): This metric is the average difference between the completion time and due date of all the orders. MOT is a typical metric used to indicate customer satisfaction

$$MOT = \frac{\sum_{|CO|} \max_{j\in co}(0, ct_j - dt_j)}{|CO|}.\# \tag{16}$$

- Mean *jt* maintenance cycle (MJMC): This is the average duration from when *JT* arrives to when it is completed and delivered to the customer. A shorter MJMC indicates higher resource utilization in the maintenance process.

$$MTMC = \frac{\sum_{j=1}^{|JT|}(ct_j - at_j)}{|JT|}.\# \tag{17}$$

Performance evaluation

Tables 8 and 9 present the performance of OoO and the other four comparative rules in certain and uncertain scenarios. The bold values in the table correspond to the rules that perform best in those scenarios. In a certain environment, EDD emphasizes delivery punctuality and avoids the accumulation of delays by prioritizing the jobs with the earliest due date, thus effectively reducing the MOT by the determined job information. Although OoO slightly falls behind EDD in MOT under deterministic scenarios, it performs better in uncertain environments. Moreover, OoO has the best overall performance on MS, TST, and MTMC

Table 7
Rules adopted as references in the experiments.

Rules	Type	Descriptions	Priority Values
EDD [45]	Static	To minimize the total tardiness, all the *JTs* are sorted in ascending order by their due dates. *JT* with the smallest due date is processed first.	$Prio_i = \frac{1}{dt_i}$
SPT [46]	Static	To minimize inefficient use of maintenance resources, all the *JTs* are sorted in ascending order by their standard processing time. *JT* with the smallest processing time is processed first.	$Prio_i = \frac{1}{spt_i}$
SST [47]	Dynamic	Initialize all *JTs* sorted on a first-come-first-served principle. To minimize the total setup time, once completing jt_{i-1}, search for the shortest setup time jt_i in the remaining *JTQ* until all jobs are completed.	$Prio_i = \frac{1}{st^h_{i-1,i}}$
JIT [47]	Dynamic	To complete the task delivery at the right time to meet the customer demand, all the *JTs* are released and executed as closely to their due dates as possible.	$Prio_i = (dt_i - spt_i - \tau)^2$

across all the scenarios.

Tables 10 and 11 illustrate the extent to which OoO outperforms other rules across varying order volumes and uncertainties. The comparison results indicate that the order volume has a minimal impact on the performance of OoO when environmental factors remain constant. This implies that OoO can be widely extended and applied to scenarios with any order volume. Furthermore, the overall performance of OoO exhibits a slight enhancement as the order volume increased. For example, in a certain scenario, increasing the order quantity from 30 to 100 results in OoO showing an average improvement of 2 % compared with the other four rules. Additionally, when the environment transitions from certain to uncertain while maintaining the same number of orders, the improvement in OoO remains steady, with slight enhancements observed. Specifically, the effectiveness of OoO in reducing order tardiness significantly increases from an average of 21–32 %. In an uncertain environment, OoO shows a massive average reduction of approximately 56 % in TST and moderate average improvements of approximately 23 % in MS, 32 % in MOT, and 23 % in MTMC.

Fig. 7 and Fig. 8 provide intuitive descriptions of the box plots used to measure the performance across the six scenarios for the 5 rules. Consistent with the results shown in Tables 8–11, OoO performs as an optimal rule in both certain and uncertain environments. Fig. 7 and Fig. 8 show that the box lengths of OoO are shorter than those of the other methods, which suggests that the performance of OoO is more stable than that of EDD, SPT, SST, and JIT, regardless of the total number of orders and uncertainty level. In addition to OoO, EDD effectively reduces lateness rates, SST is excellent at reducing the total setup time, and SPT has a certain effect on decreasing the aircraft holding time in hangars.

Table 8
Performance evaluations in certain scenarios.

	OoO	**EDD**	**SPT**	**SST**	**JIT**
30 Orders					
MS	**10115**	13025	13001	12528	13175
TST	**956**	2331	2316	1785	2330
MOT	5172	**4679**	7123	7382	7616
MTMC	**4946**	6284	5830	6432	6788
50 Orders					
MS	**16289**	21525	21533	20314	21341
TST	**1597**	3929	3911	2975	3906
MOT	8545	**7923**	12015	12235	12686
MTMC	**7999**	10304	9595	10454	10968
100 Orders					
MS	**32841**	43213	43182	40932	42951
TST	**3243**	7940	7905	5988	7910
MOT	17625	**16331**	24674	25236	26254
MTMC	**16143**	20691	19212	21157	22167

Table 9
Performance evaluations in uncertain scenarios.

	OoO	EDD	SPT	SST	JIT
30 Orders					
MS	**14828**	19238	19229	18463	19384
TST	**1411**	3517	3510	2671	3508
MOT	**6638**	7043	10727	11099	11536
MTMC	**7184**	9231	8595	9515	9974
50 Orders					
MS	**24303**	31828	31824	30300	31825
TST	**2430**	5876	5873	4458	5889
MOT	**11165**	11902	18066	18677	19380
MTMC	**11777**	15208	14158	15711	16476
100 Orders					
MS	**48873**	64159	64149	60988	64008
TST	**4904**	11951	11938	8990	11926
MOT	**22910**	24444	36973	37872	39436
MTMC	**23781**	30631	29321	31597	33083

To further quantify the advantage of OoO in uncertain scenarios, we compare the degree of degradation for each rule when transitioning from certain to uncertain scenarios with the same order volume. Grouping bar charts are plotted to illustrate the relative degradation rates of the comparison rules with OoO (shown in Fig. 9). The occurrence of uncertain events adversely affects the performance of all the rules. However, the degradation of OoO is significantly smaller than that of the other four rules, which suggests that OoO is the most robust rule among them. Fig. 9 shows that, under uncertain environments, the degradation levels of OoO for MS and MTMC remain approximately 30 % and 38 % lower than those of the other rules, respectively. Additionally, OoO and EDD exhibit less degradation in enhancing order delivery punctuality, whereas OoO and SST demonstrate minimal degradation in reducing the total setup time. However, when faced with uncertain events, the degradation levels in MOT and TST of SPT and JIT compared with OoO exceed 140 %.

Discussion

Uncertain events such as the addition of emergency orders, resource uncertainty, and fluctuating processing time during the PSE make the maintenance process highly dynamic. From a practical perspective, it was proven that the application of OoO-AMOS is effective for providing stable PSE performance for a high-uncertain aviation maintenance company. Moreover, numerical experimental results have theoretically proven that OoO delivers excellent performance in terms of enhanced maintenance efficiency, on-time delivery, and resource utilization.

Table 10
Performance improvement of OoO compared with other rules in certain scenarios.

	Average	OoO to EDD	OoO to SPT	OoO to SST	OoO to JIT
30 Orders					
MS	**22 %**	22 %	22 %	19 %	23 %
TST	**56 %**	59 %	59 %	46 %	59 %
MOT	**20 %**	−11 %	27 %	30 %	32 %
MTMC	**22 %**	21 %	15 %	23 %	27 %
50 Orders					
MS	**23 %**	24 %	24 %	20 %	24 %
TST	**56 %**	59 %	59 %	46 %	59 %
MOT	**21 %**	−8 %	29 %	30 %	33 %
MTMC	**22 %**	22 %	17 %	23 %	27 %
100 Orders					
MS	**23 %**	24 %	24 %	20 %	24 %
TST	**56 %**	59 %	59 %	46 %	59 %
MOT	**21 %**	−8 %	29 %	30 %	33 %
MTMC	**22 %**	22 %	16 %	24 %	27 %

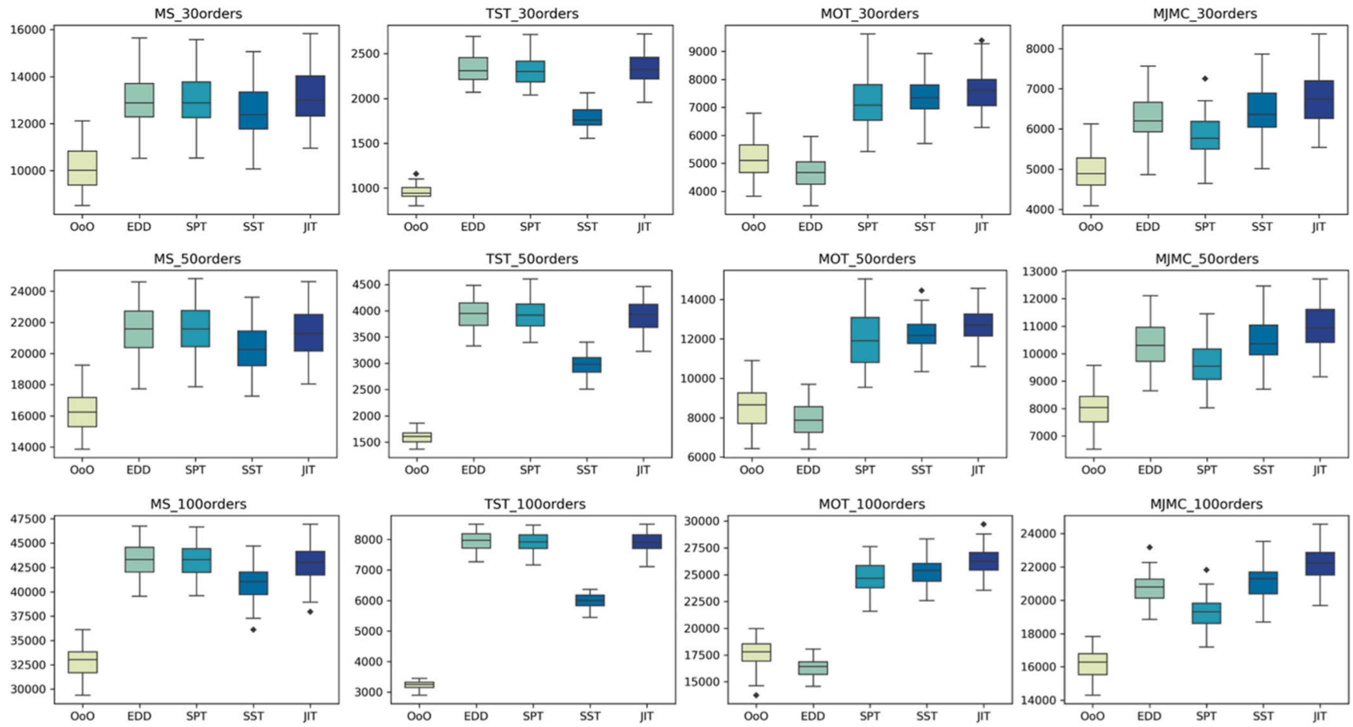

Fig. 7. Boxplots of performance measures (certain scenario).

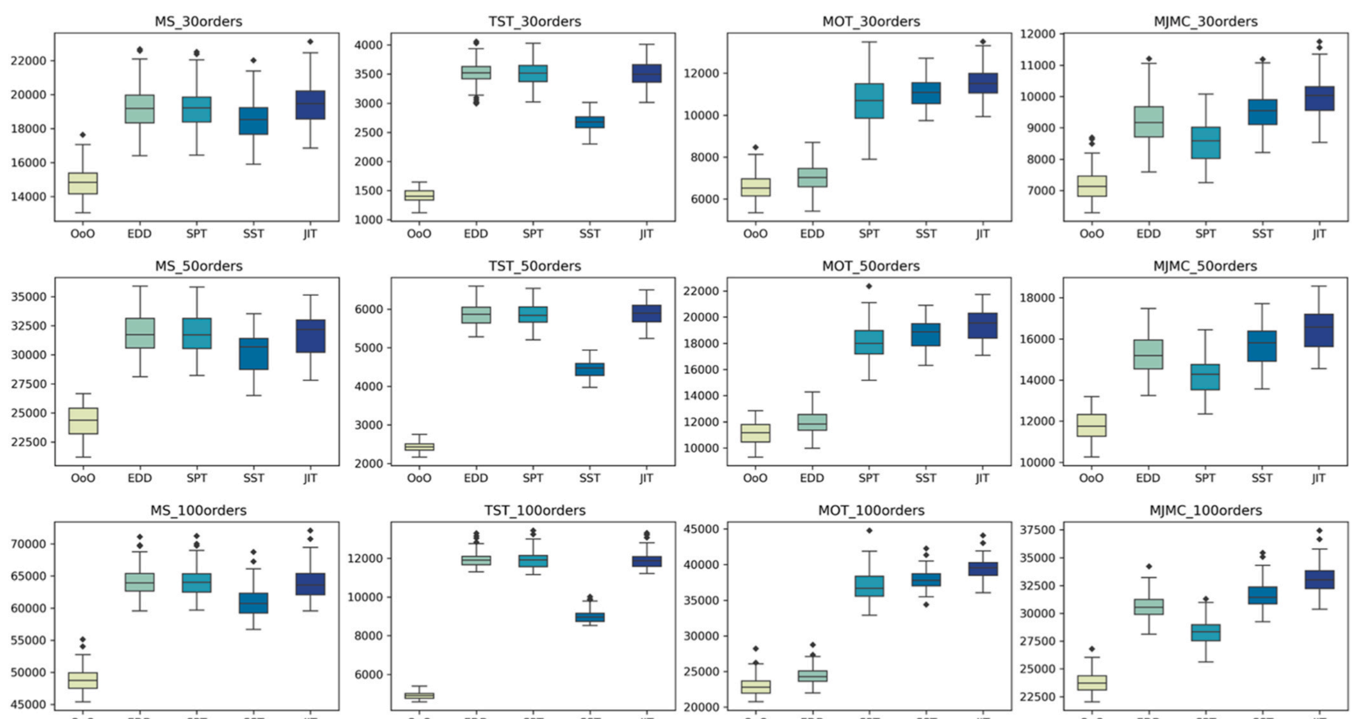

Fig. 8. Boxplots of performance measures (uncertain scenario).

Table 11
Performance improvement of OoO compared with other rules in uncertain scenarios.

	Average	OoO to EDD	OoO to SPT	OoO to SST	OoO to JIT
30 Orders					
MS	**22 %**	23 %	23 %	20 %	24 %
TST	**57 %**	60 %	60 %	47 %	60 %
MOT	**32 %**	6 %	38 %	40 %	42 %
MTMC	**23 %**	22 %	16 %	24 %	28 %
50 Orders					
MS	**23 %**	24 %	24 %	20 %	24 %
TST	**55 %**	59 %	59 %	45 %	59 %
MOT	**32 %**	6 %	38 %	40 %	42 %
MTMC	**23 %**	23 %	17 %	25 %	29 %
100 Orders					
MS	**23 %**	24 %	24 %	20 %	24 %
TST	**56 %**	59 %	59 %	45 %	59 %
MOT	**31 %**	6 %	38 %	40 %	42 %
MTMC	**24 %**	22 %	19 %	25 %	28 %

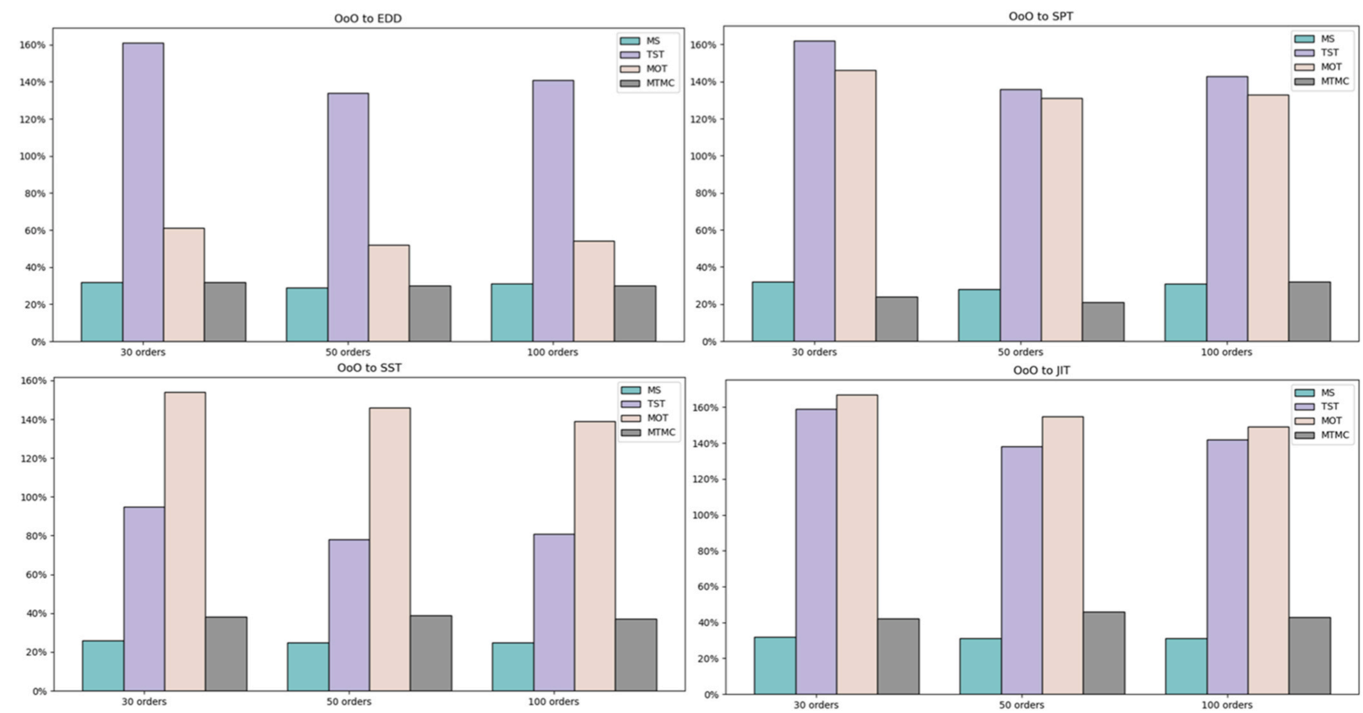

Fig. 9. Relative degradation of performance indicators under uncertainty.

Specifically, the total setup time was reduced by more than 55 %, highlighting the efficiency gains achieved through minimizing the unnecessary setup and waiting time. Additionally, significant improvements were observed in other key performance metrics: maintenance makespan decreased by more than 22 %, average order tardiness was reduced by more than 31 %, and the hangar turnover rate improved by more than 23 %. These results underscore the impact of OoO in enhancing operational efficiency and resource utilization in aviation maintenance environments. By utilizing real-time on-site data, OoO is capable of making fine-grained decisions during the execution phase, which reduces the impact of unpredictable and uncertain events in the execution phase. Therefore, the proposed OoO-AMOS is suitable and effective for minimizing the influence caused by uncertainties in the multistage maintenance process.

Conclusions

This research proposes OoO-AMOS as an integrated solution for reducing the influence of uncertainty on aviation maintenance PSE. Firstly, advanced technologies were employed to manage maintenance resources intelligently and capture uncertainties hidden in PSE at the same time. Afterward, the collected real-time on-site data are used as inputs to an OoO-based thread manager for real-time decision-making under the PSE phase. Moreover, a resource manager and an FSM-based validation mechanism were deployed to formulize the state transitions of multiple maintenance resources. Finally, a real-life case study was conducted with our research collaborators to demonstrate the effectiveness of OoO-AMOS from both theoretical and practical sides. The numerical experiments evaluated the performance of the OoO-enabled mechanism in terms of maintenance efficiency, punctuality, and resource utilization. The results revealed that the adoption of the OoO-based decision-making mechanism in the uncertain scenario reduced the total setup time by more than 55 %, whereas the maintenance makespan, average order tardiness, and hangar turnover rates improved by more than 22 %, 31 %, and 23 %, respectively.

The main contributions of this paper are as follows. First, this study is the first attempt at investigating the underlying logic of the OS for handling the uncertainty in aviation maintenance. Second, the framework of aviation maintenance PSE processes was constructed through a newly designed thread manager. The OoO-enabled decision-making mechanism is employed as the theoretical foundation for the thread manager, thereby breaking through the traditional sequence mechanism and mitigating the impact of uncertainty on the maintenance process. Third, the integration of the resource manager and FSM-based operation validation mechanism enables aviation maintenance companies to effectively manage and analyze digitized resource data, thereby realizing the value of real-time information in maintenance scenarios.

However, this study has some limitations. OoO-AMOS strongly relies on real-time data. Therefore, if there are any errors in data collection due to factors such as improper deployment of smart devices in real-world scenarios, the performance of OoO-AMOS will be greatly affected. In addition, when using OoO-AMOS in different aviation maintenance companies, the weights of work priorities need to be rationally adjusted according to the actual situation in the field and the expectations of the company management. Future research will consider more complex maintenance constraints and objectives.

CRediT authorship contribution statement

Yang Fan: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Liu Wei:** Writing – review & editing, Supervision. **Ren Cheng:** Writing – review & editing, Methodology. **Li Ming:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Li Mingxing:** Writing – review & editing, Methodology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This paper is partially supported by three grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU15208824, C7076-22G and T32-707/22-N) and the Innovation and Technology Commission of the HKSAR Government through the InnoHK initiative.

References

[1] Ogunfowora O, Najjaran H. Reinforcement and deep reinforcement learning-based solutions for machine maintenance planning, scheduling policies, and optimization. J Manuf Syst 2023;70:244–63. https://doi.org/10.1016/j.jmsy.2023.07.014.
[2] H.A. Kinnison and T. Siddiqui, “Aviation maintenance management,” 2013, Accessed: Dec. 20, 2024. [Online]. Available: https://commons.erau.edu/cgi/viewcontent.cgi?article= 2668&context=publication.
[3] Li T, Lockett H, Lawson C. Using requirement-functional-logical-physical models to support early assembly process planning for complex aircraft systems integration. J Manuf Syst 2020;54:242–57. https://doi.org/10.1016/j.jmsy.2020.01.001.
[4] Cooper T, Smiley J, Porter C, Precourt C. Global Fleet & MRO Market Forecast Commentary. Olyver Wyman; 2018.
[5] Jiang Y, et al. Digital twin-enabled real-time synchronization for planning, scheduling, and execution in precast on-site assembly. Autom Constr 2022;141:104397. https://doi.org/10.1016/j.autcon.2022.104397.
[6] Li M, Li M, Ding H, Ling S, Huang GQ. Graduation-inspired synchronization for industry 4.0 planning, scheduling, and execution. J Manuf Syst 2022;64:94–106. https://doi.org/10.1016/j.jmsy.2022.05.017.
[7] Masmoudi M, Haït A. Fuzzy uncertainty modelling for project planning: application to helicopter maintenance. Int J Prod Res 2012;50(13):3594–611. https://doi.org/10.1080/00207543.2012.670925.
[8] Niu B, Xue B, Zhong H, Qiu H, Zhou T. Short-term aviation maintenance technician scheduling based on dynamic task disassembly mechanism. Inf Sci Jun. 2023;629:816–35. https://doi.org/10.1016/j.ins.2023.01.137.
[9] Li M, Li M, Guo D, Qu T, Huang GQ. Real-time data-driven out-of-order synchronization for production and intralogistics in multiresource-constrained assembly systems. IEEE Trans Syst Man Cyber, Syst 2023:1–13. https://doi.org/10.1109/TSMC.2023.3298927.
[10] Deng Q, Santos BF, Curran R. A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. Eur J Oper Res 2020;281(2):256–73. https://doi.org/10.1016/j.ejor.2019.08.025.
[11] Dinis D, Barbosa-Póvoa A, Teixeira ÂP. A supporting framework for maintenance capacity planning and scheduling: Development and application in the aircraft MRO industry. Int J Prod Econ 2019;218:1–15. https://doi.org/10.1016/j.ijpe.2019.04.029.
[12] Samaranayake P, Kiridena S. Aircraft maintenance planning and scheduling: an integrated framework. J Qual Maint Eng 2012;18(4):432–53. https://doi.org/10.1108/13552511211281598.
[13] Heik D, Bahrpeyma F, Reichelt D. Study on the application of single-agent and multi-agent reinforcement learning to dynamic scheduling in manufacturing environments with growing complexity: Case study on the synthesis of an industrial IoT Test Bed. J Manuf Syst 2024;77:525–57. https://doi.org/10.1016/j.jmsy.2024.09.019.
[14] Tanenbaum AS, Woodhull AS. Operating Systems: Design and Implementation. Prentice Hall Englewood Cliffs; 1997. Vol. 68.
[15] Yang X, He Y, Liao R, Cai Y, Dai W. Mission reliability-centered opportunistic maintenance approach for multistate manufacturing systems. Reliab Eng Syst Saf 2024;241:109693. https://doi.org/10.1016/j.ress.2023.109693.
[16] M. Flynn, “Computer Architecture,” in Wiley Encyclopedia of Computer Science and Engineering, John Wiley & Sons, Ltd, 2007. doi: 10.1002/9780470050118.ecse071.
[17] Boere NJ. Air Canada Saves with Aircraft Maintenance Scheduling. Interfaces 1977;7(3):1–13. https://doi.org/10.1287/inte.7.3.1.
[18] Ben Ahmed M, Hryhoryeva M, Hvattum LM, Haouari M. A matheuristic for the robust integrated airline fleet assignment, aircraft routing, and crew pairing problem. Comput Oper Res 2022;137:105551. https://doi.org/10.1016/j.cor.2021.105551.
[19] Erkoc M, Ertogral K. Overhaul planning and exchange scheduling for maintenance services with rotable inventory and limited processing capacity. Comput Ind Eng 2016;98:30–9. https://doi.org/10.1016/j.cie.2016.05.021.
[20] Hu Y, Miao X, Zhang J, Liu J, Pan E. Reinforcement learning-driven maintenance strategy: a novel solution for long-term aircraft maintenance decision optimization. Comput Ind Eng 2021;153:107056. https://doi.org/10.1016/j.cie.2020.107056.
[21] Başdere M, Bilge Ü. Operational aircraft maintenance routing problem with remaining time consideration. Eur J Oper Res 2014;235(1):315–28. https://doi.org/10.1016/j.ejor.2013.10.066.
[22] Chen Q, Li M, Xu G, Huang GQ. Cyber-physical spare parts intralogistics system for aviation MRO. Adv Eng Inform 2023;56:101919. https://doi.org/10.1016/j.aei.2023.101919.
[23] Qin Y, Wang ZX, Chan FTS, Chung SH, Qu T. A mathematical model and algorithms for the aircraft hangar maintenance scheduling problem. Appl Math Model 2019;67:491–509. https://doi.org/10.1016/j.apm.2018.11.008.
[24] Deng Q, Santos BF, Verhagen WJC. A novel decision support system for optimizing aircraft maintenance check schedule and task allocation. Decis Support Syst 2021;146:113545. https://doi.org/10.1016/j.dss.2021.113545.
[25] Weide TVD, Deng Q, Santos BF. Robust long-term aircraft heavy maintenance check scheduling optimization under uncertainty. Comput Oper Res 2022;141:105667. https://doi.org/10.1016/j.cor.2021.105667.
[26] Li M, Zhong RY, Qu T, Huang GQ. Spatial–temporal out-of-order execution for advanced planning and scheduling in cyber-physical factories. J Intell Manuf 2022;33(5):1355–72. https://doi.org/10.1007/s10845-020-01727-2.
[27] Ding H, Li M, Zhong RY, Huang GQ. Multistage self-adaptive decision-making mechanism for prefabricated building modules with IoT-enabled graduation manufacturing system. Autom Constr 2023;148:104755. https://doi.org/10.1016/j.autcon.2023.104755.
[28] Sun M, Ding J, Zhao Z, Chen J, Huang GQ, Wang L. “Out-of-order execution enabled deep reinforcement learning for dynamic additive manufacturing scheduling,”. Robot Comput-Integr Manuf 2025;91:102841. https://doi.org/10.1016/j.rcim.2024.102841.
[29] Papakostas N, Papachatzakis P, Xanthakis V, Mourtzis D, Chryssolouris G. An approach to operational aircraft maintenance planning. Decis Support Syst 2010;48(4):604–12. https://doi.org/10.1016/j.dss.2009.11.010.
[30] Zhao X, Deng Q, Liu X, Zhang L, Wu S, Jiang C. Integrated scheduling of distributed service resources for complex equipment considering multiple on-site MRO tasks. Int J Prod Res 2022;60(10):3219–36. https://doi.org/10.1080/00207543.2021.1916117.
[31] Yang T-H, Yan S, Chen H-H. An airline maintenance manpower planning model with flexible strategies. J Air Transp Manag 2003;9(4):233–9. https://doi.org/10.1016/S0969-6997(03)00013-9.
[32] Zhang Q, Zheng S, Yu C, Wang Q, Ke Y. Digital thread-based modeling of digital twin framework for the aircraft assembly system. J Manuf Syst 2022;65:406–20. https://doi.org/10.1016/j.jmsy.2022.10.004.
[33] Fang X, Song Q, Li Z, Wang X, Ma H, Liu Z. Intelligent monitoring system for production lines in smart factories: a hybrid method integrating Transformer and Kalman filter. J Manuf Syst 2025;79:27–47. https://doi.org/10.1016/j.jmsy.2024.12.014.
[34] Guo D, Li M, Zhong R, Huang GQ. Graduation Intelligent Manufacturing System (GiMS): an Industry 4.0 paradigm for production and operations management. IMDS 2020;121(1):86–98. https://doi.org/10.1108/IMDS-08-2020-0489.
[35] Lin P, Shen L, Zhao Z, Huang GQ. Graduation manufacturing system: synchronization with IoT-enabled smart tickets. J Intell Manuf Dec. 2019;30(8):2885–900. https://doi.org/10.1007/s10845-018-1429-4.
[36] Milligan GW, Cooper MC. A study of standardization of variables in cluster analysis. J Classif 1988;5:181–204.

[37] Nascimento MC, De Carvalho AC. Spectral methods for gragh clustering-a survey. Eur J Oper Res 2011;211(2):221–31.
[38] Von Luxburg U. A tutorial on spectral clustering. Stat Comput 2007;17(4): 395–416. https://doi.org/10.1007/s11222-007-9033-z.
[39] Lee D, Yannakakis M. Principles and methods of testing finite state machines-a survey. Proc IEEE 1996;84(8):1090–123. https://doi.org/10.1109/5.533956.
[40] Gui Y, Tang D, Zhu H, Zhang Y, Zhang Z. Dynamic scheduling for flexible job shop using a deep reinforcement learning approach. Comput Ind Eng 2023;180:109255. https://doi.org/10.1016/j.cie.2023.109255.
[41] Jing H, Sheng L, Luo C, Kwak C. Statistical analysis of family based dispatching rules and preemption. Int J Prod Econ 2021;240:108242. https://doi.org/10.1016/j.ijpe.2021.108242.
[42] 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;61:423–49. https://doi.org/10.1016/j.jmsy.2021.09.018.
[43] Pinedo ML. Scheduling. Cham: Springer International Publishing; 2016. ⟨10.1007/978-3-319-26580-3⟩.
[44] Đurasević M, Jakobović D. A survey of dispatching rules for the dynamic unrelated machines environment. Expert Syst Appl 2018;113:555–69. https://doi.org/10.1016/j.eswa.2018.06.053.
[45] Bryant R, Lakner P, Pinedo M. On the optimality of the earliest due date rule in stochastic scheduling and in queueing. Eur J Oper Res 2022;298(1):202–12. https://doi.org/10.1016/j.ejor.2021.09.039.
[46] Rose O. The shortest processing time first (SPTF) dispatch rule and some variants in semiconductor manufacturing. *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*. Arlington, VA, USA: IEEE; 2001. p. 1220–4. ⟨10.1109/WSC.2001.977437⟩.
[47] Li M, Guo D, Li M, Qu T, Huang GQ. Operation twins: production-intralogistics synchronisation in Industry 4.0. Int J Prod Res 2023;61(15):5193–211. https://doi.org/10.1080/00207543.2022.2098874.