Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.069492

ARTICLE

A Q-Learning Improved Particle Swarm Optimization for Aircraft Pulsating Assembly Line Scheduling Problem Considering Skilled Operator Allocation

Xiaoyu Wen[1,2], Haohao Liu[1,2], Xinyu Zhang[1,2], Haoqi Wang[1,2], Yuyan Zhang[1,2], Guoyong Ye[1,2], Hongwen Xing[3], Siren Liu[3] and Hao Li[1,2,*]

[1]Henan Provincial Key Laboratory of Intelligent Manufacturing of Mechanical Equipment, Zhengzhou University of Light Industry, Zhengzhou, 450000, China
[2]School of Mechanical and Electrical Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450000, China
[3]COMAC Shanghai Aircraft Manufacturing Co., Ltd., Shanghai, 200000, China

*Corresponding Author: Hao Li. Email: lihao@zzuli.edu.cn

Received: 24 June 2025; Accepted: 04 September 2025; Published: 10 November 2025

ABSTRACT: Aircraft assembly is characterized by stringent precedence constraints, limited resource availability, spatial restrictions, and a high degree of manual intervention. These factors lead to considerable variability in operator workloads and significantly increase the complexity of scheduling. To address this challenge, this study investigates the Aircraft Pulsating Assembly Line Scheduling Problem (APALSP) under skilled operator allocation, with the objective of minimizing assembly completion time. A mathematical model considering skilled operator allocation is developed, and a Q-Learning improved Particle Swarm Optimization algorithm (QLPSO) is proposed. In the algorithm design, a reverse scheduling strategy is adopted to effectively manage large-scale precedence constraints. Moreover, a reverse sequence encoding method is introduced to generate operation sequences, while a time decoding mechanism is employed to determine completion times. The problem is further reformulated as a Markov Decision Process (MDP) with explicitly defined state and action spaces. Within QLPSO, the Q-learning mechanism adaptively adjusts inertia weights and learning factors, thereby achieving a balance between exploration capability and convergence performance. To validate the effectiveness of the proposed approach, extensive computational experiments are conducted on benchmark instances of different scales, including small, medium, large, and ultra-large cases. The results demonstrate that QLPSO consistently delivers stable and high-quality solutions across all scenarios. In ultra-large-scale instances, it improves the best solution by 25.2% compared with the Genetic Algorithm (GA) and enhances the average solution by 16.9% over the Q-learning algorithm, showing clear advantages over the comparative methods. These findings not only confirm the effectiveness of the proposed algorithm but also provide valuable theoretical references and practical guidance for the intelligent scheduling optimization of aircraft pulsating assembly lines.

KEYWORDS: Aircraft pulsating assembly lines; skilled operator; reinforcement learning; PSO; reverse scheduling

1 Introduction

In the manufacturing of large aircraft, assembly operations account for approximately 50%~70% of total direct manufacturing costs, excluding production preparation and industrial equipment fabrication [1]. Automated and flexible assembly techniques have long been central to advances in aircraft production. Sub-assembly in aircraft manufacturing focuses on "partial manufacturing," whereas final assembly represents the "integration of the entire aircraft". The aircraft pulsating final assembly line is divided into stations according to the assembly outline, enabling the aircraft to move between stations with an inherent takt time, thereby

improving assembly efficiency and shortening the production cycle. Among these, final assembly plays a pivotal role in determining both the production rate and overall product quality [2].

With growing demand for commercial, military, and general-aviation aircraft in China, developing assembly lines that are more cost-effective, efficient, and capable of ensuring high quality has become a strategic priority. Traditional fixed hangar-style assembly reveals significant limitations when confronted with increasingly complex aircraft designs and rising production volumes. Drawing on successful practices in the automotive industry, the aviation sector has begun to adopt mobile assembly methods. These are commonly categorized by automation level into "pulsating assembly lines" and "continuously moving assembly lines" [3]. In a pulsating assembly line, the assembly outline (AO, the smallest unit of task assignment in the production system) is distributed across a limited number of workstations. Each station is assigned a set of tasks to be completed within a fixed cycle; at the end of the cycle, the aircraft advances to the next station, yielding a stepwise, pulse-like movement. By contrast, a continuously moving assembly line operates with higher automation and a stable material supply: the aircraft moves at a constant speed while operators complete tasks sequentially. At present, aircraft pulse assembly lines are mainly constrained by multiple factors, including material supply, precedence relations among tasks, resource availability, spatial limitations, and operator collaboration. Materials are supplied by multiple vendors, and large commercial aircraft involve numerous and complex operational constraints. The resources primarily include tooling equipment and ordinary operator resources. Each operator occupies space within the aircraft section while working, and the execution of assembly tasks often requires the cooperation of multiple types and numbers of workers. Therefore, pulse final assembly lines have attracted extensive academic attention worldwide.

The workstation segmentation inherent to aircraft pulsating assembly lines gives rise to two major research topics: the "Aircraft Pulsating Assembly Line Balancing Problem (APALBP)" and the "Aircraft Pulsating Assembly Line Scheduling Problem (APALSP)". The APALBP [4] focuses on distributing assembly tasks across stations to balance workload and minimize idle time. This problem has been widely studied in aircraft assembly and other manufacturing domains. For example, Bao et al. [5] developed an integer linear programming model and proposed a two-stage heuristic for balancing an aircraft final assembly line. Korivand et al. [6] introduced a framework for predicting human performance in human-robot teaming by integrating physiological data with Q-learning, achieving 95.45% accuracy; their findings offer insights relevant to resource and performance optimization in aircraft assembly. In contrast, the APALSP [7] addresses the allocation of resources and operators under spatial constraints, as well as the scheduling of task start and finish times to ensure timely assembly completion. The resources considered include various tool resources and different types of operators, each of whom occupies space within the aircraft sections. Owing to its diverse focus areas, APALSP has spawned several subproblems. Prior studies have examined related topics such as multi-operator parallel scheduling [8], multi-skilled operator allocation [9,10], multi-level operator assignment [11], scheduling with operator fatigue [12], workload-fluctuation scheduling [13], and human-robot collaborative scheduling [14]. The present research concentrates on one such subproblem: the aircraft pulsating assembly line scheduling problem with skilled operator allocation. The aircraft assembly process is subject to strict task sequencing constraints, and skilled workers impact the completion time of each task. Therefore, efficient operator scheduling [15] and task scheduling are crucial for improving assembly efficiency and reducing production costs.

Building on this literature, many studies emphasize operator allocation yet often treat operators as homogeneous resources. In practice, however, training skilled operators requires substantial investment, and their deployment directly affects assembly efficiency, product quality, and production stability. Skilled operators execute complex tasks more accurately and efficiently. Therefore, explicitly modeling skilled operators as a distinct resource within the scheduling process is both critical and challenging.

The remainder of the article is organized as follows. Section 2 reviews APALSP from the perspectives of resource constraints and operator allocation. Section 3 details the problem statement and mathematical formulation. Section 4 presents the Q-Learning improved Particle Swarm Optimization (QLPSO) approach. Section 5 reports experimental validation on small-, medium-, large-, and ultra-large-scale test cases. Section 6 concludes the study and outlines avenues for future research.

2 Research Overview

The aircraft pulsating assembly line scheduling problem can be regarded as a variant of the project scheduling problem with unique constraints. It extends and generalizes the classical Resource-Constrained Project Scheduling Problem (RCPSP) [16]. As a key area in operations research, RCPSP has been extensively studied and is considered a mature topic in both modeling and algorithm development. Li et al. [17] proposed a proactive scheduling model for RCPSP, incorporating resource constraints, and solved it using a branch-and-bound algorithm. Their method outperformed CPLEX and traditional genetic algorithms. Liu et al. [18] introduced a bi-objective model for finance-based and resource-constrained robust project scheduling (FBRCRPSP). They developed an enhanced non-dominated sorting genetic algorithm (NSGA-II-LS), which efficiently explores the neighborhood space and performs well on small-scale instances. Currently, research in this area continues to evolve, providing valuable insights and decision-making support for project managers.

The Aircraft Pulsating Assembly Line Scheduling Problem (APALSP) remains in the early stages of academic exploration. This is largely due to the relatively late development of China's aircraft manufacturing and assembly industry, where research across multiple dimensions is still emerging. Roussel et al. [19] conducted preliminary studies on design optimization for aircraft assembly lines. Traditionally, manufacturing systems are developed after the aircraft design is finalized, often resulting in suboptimal performance. To address this, the authors proposed integrating production system considerations into the early stages of aircraft design. They introduced a constraint programming encoding and an ε-constraint-based algorithm, which demonstrated significant potential. In parallel, Long et al. [20] investigated productivity prediction for aircraft final assembly lines. Their study compared three modeling approaches: a simulation-based prediction model, a representative regression model, and a machine learning model including multilayer perception, gradient boost regression tree and random forest. These methods were validated using a real-world final assembly line involving three types of aircraft, confirming their practical applicability. Both domestic and international scholars continue to focus on simulation and optimization studies of aircraft final assembly pulsating lines. These efforts aim to improve resource utilization and achieve better operational balance across production stages.

Research on the Aircraft Pulsating Assembly Line Scheduling Problem (APALSP) primarily focuses on resource constraints, as this is the most classical and fundamental modeling framework. Many researchers have conducted studies based on resource-constrained models, considering various tooling resources and operator resources, and some scholars have also taken spatial constraints into account. Cai et al. [21] addressed spatial constraints in the aircraft assembly process by formulating a mathematical model for sub-assembly scheduling. Their objective was to minimize total assembly duration while accounting for spatial limitations. To solve the model, they proposed an Improved Genetic Algorithm with Variable Neighborhood Search (IGA-VNS). Similarly, Wen et al. [7] applied a reinforcement learning-based genetic algorithm (QIGA) to handle multiple constraints in aircraft assembly scheduling. By integrating genetic algorithms with Markov decision models and dynamically adjusting parameters, the method enhanced the search efficiency. Experimental results demonstrated that the proposed algorithm significantly reduced total assembly time in large-scale aircraft assembly scenarios. Shan et al. [22] also focused on resource constraints but treated them

as an objective function within the scheduling problem. They developed a comprehensive fitness function to evaluate resource balance in demand-driven scheduling and introduced an Adaptive Genetic Algorithm (GA) to address the problem. Further research has delved deeper into flexible resource allocation. Ren et al. [23] proposed a flexible resource investment model based on project splitting (FRIP_PS), which allows for dynamic resource allocation in final assembly scheduling. They embedded a heuristic method into a genetic algorithm to solve the problem efficiently. In subsequent work, Ren et al. [24] incorporated resource transfer times between stations into the scheduling model. Considering task precedence and resource limitations, they proposed a linear mathematical model aimed at minimizing total project completion time. A branch-and-bound embedded genetic algorithm was developed to solve this model. To address the challenges of multi-workstation parallel execution and the unavailability of line-end shared resources, Lu et al. [25] introduced a new variant of the scheduling problem: the Resource Investment Problem based on Project Splitting with Time Windows (RIPPS_TW). This model allows projects to be divided into sub-projects, enhancing scheduling flexibility. They also proposed a two-stage iterative loop algorithm and validated its effectiveness through experimental case studies. Borreguero et al. [26] developed a multi-mode resource-constrained project scheduling model that incorporates incompatibility constraints, resource limitations, and a large task volume. Using a constraint programming-based large neighborhood search algorithm, they achieved significantly better results compared to the company's previously applied heuristics.

Focusing on resource constraints, both field research and literature reviews reveal an increasing emphasis on the allocation and arrangement of assembly line personnel in aircraft production. As a result, many scholars have explored workforce allocation in greater depth. Xin et al. [9] addressed the scheduling problem of moving production lines in aircraft assembly. They considered scenarios where multiple operators could jointly execute a task and developed a model for scheduling multi-operator parallel tasks. Their goal was to minimize total assembly completion time. They compared various heuristic algorithms with different rule sets and designed a genetic algorithm (GA) for performance benchmarking. Fang et al. [27] examined the allocation of workers across multi-level stations. Since certain specialized operators must shift between stations at fixed intervals, the objective was to minimize cycle time while balancing workloads. To tackle this, an improved non-dominated sorting genetic algorithm (NSGA-IV) was proposed. Some researchers take a more human-centered perspective. Arkhipov et al. [28] studied task allocation under both economic and ergonomic constraints, aiming to optimize task schedules. They proposed two models and validated them through a case study. Ottogalli et al. [29] used virtual reality (VR) to assess automated and semi-automated cargo handling processes, selecting the most efficient one. They also analyzed operator ergonomics in tight fuselage spaces, focusing on safe human-robot collaboration. Their results showed notable reductions in assembly time and labor costs. Human-robot collaboration has become a key research area. Wang et al. [14] emphasized that operator movements across stations, especially when skills differ, significantly affect final assembly efficiency. He introduced the dynamic worker allocation problem for aircraft final assembly (DWA-AFA) and proposed a human-machine collaborative optimization method (HMC-O). Tereshchuk et al. [30] developed a scheduling method for aircraft assembly involving multiple collaborative robots. They addressed the soft task priority constraint, which arises from tasks requiring different tools. Their multi-robot task allocation model reduced tool change time by incorporating priority constraints. A two-stage, data-driven approach was proposed to select task priorities, along with a soft-priority iterative auction strategy and machine learning-based scheduling. Guo [31] focused on coordinating tasks between human and robot teams (HRTs). He proposed a hybrid approach combining mixed-integer linear programming (MILP) and constraint programming (CP). The CP method outperformed others in solving real-world scheduling challenges in synchronized aircraft production and logistics.

Existing research on aircraft assembly line scheduling under resource constraints has provided a solid theoretical and algorithmic foundation. However, the critical role of operator allocation has not been fully addressed. Effective operator assignment can significantly enhance both the efficiency and quality of the assembly process. While numerous studies have examined scheduling and workforce allocation in aircraft assembly, further investigation is needed in areas such as operator allocation, multi-objective optimization, and dynamic scheduling. Building on this gap, the paper proposes a scheduling model for the Aircraft Pulsating Assembly Line Scheduling Problem considering skilled operator allocation, and design the QLPSO algorithm to solve it. This work aims to offer both theoretical insights and practical guidance for improving production efficiency in the manufacturing industry.

3 Problem Description and Formulation

3.1 Problem Description

An aircraft pulse assembly line refers to the process in which aircraft components enter the assembly workshop to complete all assembly tasks and testing until the completion of the test flight. An aircraft involves a large number of AO, each of which must strictly follow its precedence constraints. The execution of AO also requires specific resources, including various tooling resources and multiple operators. During task execution, operators occupy the space within the aircraft section, and the available space varies across different sections of the aircraft. Therefore, operator assignment must also strictly adhere to spatial constraints. These constraints affect total assembly time, restrict the number of operators, and make workforce allocation critical, as operator differences can significantly influence task durations.

Therefore, solving such scheduling problems requires considering multiple factors, including AO precedence constraints, resource constraints, spatial constraints, and skilled worker allocation. The aircraft pulsating assembly line scheduling problem with skilled operator assignment can be defined as follows: After the aircraft's tasks are divided across stations, each station contains a set of AO. Let there be a total of J AO, referred to as AO, within a station. Each AO is subject to precedence constraints based on the Operation Sequence Plan (OSP), which can be represented using an Activity on Node (AON) diagram [32]. As illustrated in Fig. 1, the AON diagram depicts not only the sequential relationships between AO but also includes segment information, task duration, and the required number of operators for each AO.

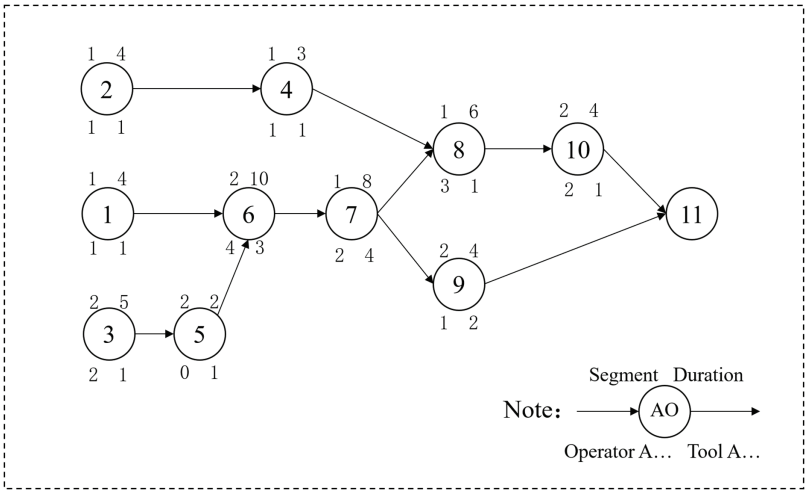


Figure 1: The activity on node (AON) for aircraft pulsating assembly line

Each segment within the assembly line is subject to a maximum capacity constraint, limiting the number of operators that can work in parallel within that segment. The duration refers to the time required to complete a specific AO, while the number of operators indicates the quantity of regular operators needed to perform the AO. Operators A and B are considered shared resources and are subject to maximum availability constraints. Skilled operators, on the other hand, can be flexibly assigned. Each AO requires at least one skilled operator to proceed under normal conditions. If two skilled operators are assigned to an AO, its duration is reduced by one-third. Assigning three skilled operators shortens the duration by half. However, allocating more than three skilled operators results in diminishing returns in efficiency, which is not considered in this study. The objective of this problem is to determine the optimal scheduling of AOs and the allocation of skilled and shared operators to minimize the total completion time.

3.2 Problem Formulation

The following symbols and their meanings are required to establish the mathematical model for the aircraft pulsating assembly line scheduling problem considering skilled operator allocation, as shown in Table 1.

Table 1: The symbols and explanations

Symbol	Explanations
T	Total completion time within the current station
M	Number of operators in the aircraft pulsating assembly line
N	Total number of AO in the aircraft pulsating assembly line
$q = \{1, 2 \dots q \dots Q\}$	Set of aircraft assembly segments, the maximum segment in the station denoted as Q
$J = \{1, 2 \dots j \dots N\}$	Set of AO identifiers
$I = \{1, 2 \dots i \dots M\}$	Set of operator identifiers
$k = \{1, 2, \dots K\}$	Resource type index
$o = \{A, B\}$	Index for regular operator types, including operator A and operator B
$S = \{1_1, 2 \dots S\}$	Set of skilled operators
$d = \{1, 2 \dots T\}$	Discrete time nodes for AOs
t_j	Duration of AO j
ST_j	Start time of AO j
ET_j	End time of AO j
P_j	Set of predecessor AOs for AO j
S_j	Set of successor AOs for AO j
R_k	Total resource available per unit time for k types of resources
r_{jk}	Resource demand for AO j for k types of resources
r_{jo}	Demand for operator type o for AO j
r_{js}	Demand for skilled operator s for AO j
e_{jq}	Demand for segment q for AO j
N_q	Space constraint for segment q, the maximum number of operators that can be accommodated
u_{jP_j}	If AO j and the previous AO P_j are completed by the same operator, the value is 1; otherwise, it is 0

(Continued)

Table 1 (continued)

Symbol	Explanations
x_{jq}	If AO j is assigned to be completed in the segment q, the value is 1; otherwise, it is 0
y_{joi}	If AO j is assigned to the operator i of type o, the value is 1; otherwise, it is 0
z_{js}	If AO j is assigned to the skilled operator s, the value is 1; otherwise, it is 0

The mathematical model for the aircraft pulsating assembly line scheduling problem considering skilled operator allocation, is as follows:

Objective:

$$\min Z = T \tag{1}$$

subject to:

$$ST_j + t_j \le T, j \in J \tag{2}$$

$$ET_j - ST_j \le t_j, j \in [1, L] \tag{3}$$

$$\sum_{j=1}^{N} x_{jq} = 1, \forall j \in [1, N] \tag{4}$$

$$ST_j - ST_{P_j} \ge t_{P_j}, j \in [1 < L] \tag{5}$$

$$ET_{P_j} - ST_j \le 0, j \in [1, L] \tag{6}$$

$$\sum_{i}^{M} x_{jq} y_{joi} < M_o, j \in [1, N], i \in [1, M], o \in [1, O], q \in Q \tag{7}$$

$$\sum_{i}^{M} x_{jq} y_{js} < M, j \in [1, N], q \in Q \tag{8}$$

$$u_{jP_j} + u_{P_j j} \ge z_{js} + z_{P_j s} - 1, j \in [1, N] \tag{9}$$

$$ST_j - ST_{P_j} \ge ST_{P_j} - M\left(1 - u_{jP_j}\right), j \in [1, N] \tag{10}$$

$$\sum x_{jq} y_{joi} r_{jk} \le R_k, j \in [1, N], i \in [1, M], o \in [1, O], k \in [1, K], q \in Q \tag{11}$$

$$\sum z_{js} \le 3 \tag{12}$$

$$\sum x_{jq} z_{js} r_{jk} \le R_k, j \in [1, N], o \in [1, O], k \in [1, K], q \in Q \tag{13}$$

$$\sum x_{jq} y_{joi} e_{jq} + x_{jq} z_{js} e_{jq} \le E, j \in [1, N], i \in [1, M], o \in [1, O], q \in Q \tag{14}$$

Objective function (1) represents the minimization of the total completion time of the aircraft assembly line. Constraint (2) represents that all AOs in the aircraft assembly line must be completed within the total completion time and cannot exceed the planned station cycle time; Constraint (3) represents that each AO must be completed within the specified time and cannot be overdue; Constraint (4) represents that each AO can only be assigned to one segment space within the station for execution; Constraints (5) and (6) represent the precedence constraints between AOs, where the start time of a succeeding AO cannot exceed the completion time of the preceding AO; Constraints (7) and (8) represent worker quantity constraints, meaning the total number of allocated operators cannot exceed the maximum number of operators available, where the former refers to regular operators and the latter refers to skilled operators; Constraints (9) and (10)

represent that at any given time, each operator can only perform one AO; Constraint (11) represents resource constraints, meaning the resource demand for parallel AOs cannot exceed the total available resources; Constraint (12) represents skilled operator constraints, meaning the number of skilled operators assigned to each AO cannot exceed three; Constraint (13) represents skilled operator resource constraints, meaning the resource demand for AOs performed by skilled operators cannot exceed the total available resources; Constraint (14) represents space constraints, meaning the space occupied by regular and skilled operators cannot exceed the maximum capacity of the segment space.

4 Q-Learning Improved Particle Swarm Optimization

Swarm Intelligence (SI), a prominent branch of Artificial Intelligence (AI), is inspired by the collective behaviors of social organisms in nature. Among various SI algorithms, Particle Swarm Optimization (PSO) stands out as one of the most widely applied and extensively studied paradigms [33]. Since its introduction in the 1990s, PSO has attracted considerable attention from both researchers and practitioners.

Despite its global search capability, PSO is prone to premature convergence in high-dimensional or complex problems and is highly sensitive to parameter settings. Reinforcement learning has been shown to alleviate these issues by adaptively tuning parameters and enhancing exploration [34]. In this study, we embed Q-learning, a model-free reinforcement learning algorithm, into PSO, enabling particles to dynamically adjust their strategies based on feedback. This integration improves convergence efficiency and increases the likelihood of attaining global or near-optimal solutions.

4.1 Design of Reverse Sequence Encoding

Due to the precedence constraints inherent in the aircraft pulsating assembly line scheduling problem, it is not feasible to generate a valid solution through purely random initialization. Instead, a priority sequence list must first be randomly generated. The value in this list represents the priority of each AO—the higher the value, the higher the AO's priority, and the earlier it should be scheduled. This priority list serves as the basis for encoding the solution. For large-scale scheduling problems with complex precedence relationships, a forward scheduling approach is typically employed. This method requires strict adherence to AO dependencies: a AO can only be executed once all of its predecessor AO have been completed. Such tight constraints greatly increase the complexity of the scheduling algorithm. To address these challenges, some researchers have explored inverse sequence-based scheduling methods. For example, Guo et al. [35] proposed a genetic algorithm based on inverse sequence virtual components to solve complex product scheduling problems with tightly coupled constraints. Similarly, Wang [36] adopted an inverse sequence scheduling strategy, which first schedules root-node processes, followed by leaf-node processes. Their experimental results demonstrate that inverse sequence scheduling offers advantages in both computational efficiency and solution quality.

As illustrated in Fig. 2, a priority sequence value is randomly assigned to each AO. When a preceding AO is completed and multiple candidate AOs are eligible for execution, the selection is made based on their priority sequence values—the AO with the higher value is scheduled first. However, this encoding strategy necessitates repeated feasibility checks. For instance, suppose AO2, AO4, and AO8 have the highest priority sequence values. According to the priority principle, AO2 should be scheduled first, followed by AO4, and then AO8. Yet, since AO8 has an uncompleted predecessor (AO7), it cannot be executed immediately. Therefore, a predecessor check is required after each AO selection to ensure precedence

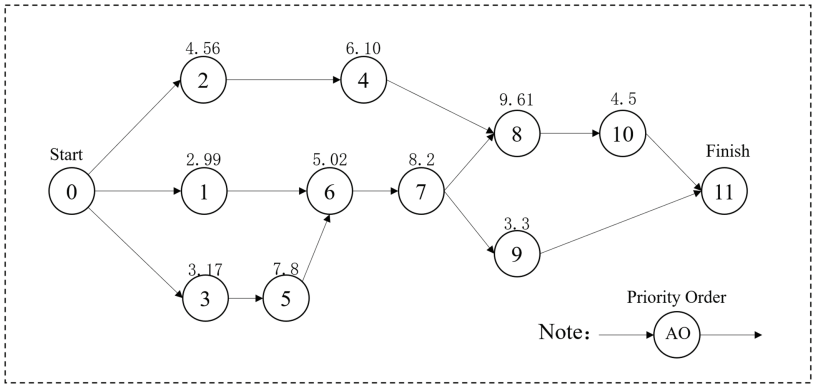


Figure 2: Sequential encoding based on priority sequence

This article introduces an inverse sequence encoding method based on AO priority. It reverses the AO precedence constraints and considers only the immediate predecessor AO for each AO. Duplicate AO in the earlier sequence is removed, retaining only the final AO order. The resulting AO sequence is then reversed again to produce a feasible schedule that satisfies all priority constraints. For example, as shown in Fig. 3, the process begins with AO11. At each step, only the AO with a higher priority value is selected as the next in sequence. This process continues iteratively, resulting in a sequence of AOs with higher priority than AO11. Duplicate AOs appearing earlier in the sequence are removed, and each unique AO is retained to form a final sequence of 11 AOs. This sequence is then reversed to produce a valid solution that adheres to the AO priority constraints.

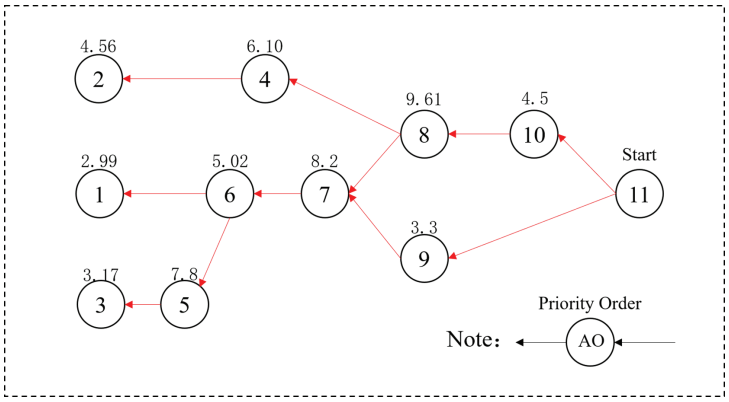


Figure 3: Reverse sequence encoding based on priority sequence

The mathematical model for the aircraft pulsating assembly line scheduling problem with skilled operator allocation employs Reverse Sequence Encoding based on priority sequences. As illustrated in Fig. 4, the encoding design consists of three layers. The first layer represents the priority sequence values, which correspond to positions in the particle swarm optimization (PSO) iteration. These values are generated randomly and used to construct the AO sequence. The second layer contains the velocity values of the particles, which are added to the priority sequence values to modify the AO sequence during the PSO update process. The third layer encodes the skilled operator assignments: one operator enables basic task execution, two operators reduce processing time by 1/3, and three operators reduce it by 1/2.

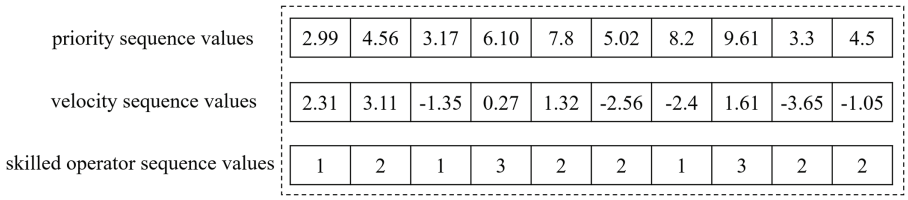


Figure 4: Encoding design of the aircraft pulsating assembly line scheduling problem

A comparative experiment was conducted using representative test cases from small-scale (20 AO), medium-scale (50 AO), large-scale (100 AO), and extra-large-scale (600 AO) AO sets. For each case, 50 independent runs of both Reverse Sequence Encoding and Sequential Encoding were executed. The solution times and results were recorded, as shown in Table 2. Both the minimum and average solution times for Reverse Sequence Encoding are consistently lower than those for Sequential Encoding. Furthermore, the time advantage of Reverse Sequence Encoding becomes more pronounced as the problem scale increases, highlighting its efficiency in handling larger instances. In terms of solution quality, Reverse Sequence Encoding also outperforms Sequential Encoding, albeit slightly. Overall, Reverse Sequence Encoding demonstrates superior algorithmic performance, particularly in obtaining near-optimal solutions within limited time. Therefore, only Reverse Sequence Encoding is employed in the experimental studies presented in Section 5.

Table 2: Comparison of results between reverse sequence encoding and sequential encoding

Problem scale	Reverse sequence encoding				Sequential encoding			
	Solution time		Solution results		Solution time		Solution results	
	Min/s	AVG/s	Min/s	AVG/s	Min/s	AVG/s	Min/s	AVG/s
20	14.49	17.10	38	42.80	16.51	19.22	40	43.42
50	49.81	54.22	146	150.60	65.61	70.53	149	151.62
100	489.16	527.80	154	163.90	573.775	603.50	157	166.00
600	16,511.8	16,641.93	540	580.30	16,920.9	16,957.19	578	621.60

4.2 Design of Time Decoding

The algorithm generates a corresponding AO sequence based on priority sequence values and calculates the completion time for that sequence. Given that AO across different sections of the aircraft pulsed assembly line can begin simultaneously, parallel execution must be considered—adding significant complexity to the decoding process. Additionally, resources are limited, so potential resource shortages must also be accounted for.

To address these challenges, the algorithm incorporates both time sequence checks and resource checks. AO are sorted based on an activity list. If both the Time Sequence Check and Resource Check are satisfied, the AO are executed in parallel. If not, the algorithm first verifies whether all predecessor AO of the current AO are completed. If not, it releases the predecessor and any incomplete AO before proceeding to the Resource Check. If available resources are insufficient, the algorithm releases and reallocates resources until the AO's requirements are met, allowing execution to resume.

Moreover, each AO requires a different number of skilled operators, which are allocated according to the skilled operator sequence values. As each operator occupies spatial resources, a Spatial Check is also incorporated. Skilled operators, along with required operators A and B, consume space within each assembly

section. Therefore, the total number of operators assigned to a AO must not exceed the section's spatial capacity. If this limit is exceeded, AO execution is delayed. This scheduling process continues until all AO are completed, ultimately determining the total assembly line completion time.

4.3 Design of QLPSO Algorithm

The PSO algorithm simulates the predatory behavior of bird flocks by iteratively updating the position and velocity of each particle. It leverages the social interactions among particles to explore the search space and find optimal solutions. However, PSO often suffers from premature convergence, frequently becoming trapped in local optima. Furthermore, its performance is highly sensitive to parameter settings; improper configuration can result in slow convergence, reduced accuracy, or stagnation in suboptimal regions. To address these issues, Tanweer et al. [37] and Han et al. [38] introduced adaptive parameter tuning methods. Tanweer proposed two strategies: one involving the self-adjustment of the inertia weight, and another employing self-awareness mechanisms to enhance global search capability. Han tackled the problem of negative knowledge transfer by evaluating the quality of transferred knowledge, thereby improving adaptability. Although these approaches improved PSO performance, the algorithm still struggles with large-scale and complex instances, such as those encountered in aircraft pulsed assembly line scheduling.

With the advancement of artificial intelligence and machine learning technologies [39–41], researchers have increasingly explored hybrid approaches combining reinforcement learning (RL) with metaheuristic algorithms. Lu et al. [42] proposed an RL-based elite ion selection strategy to guide the particle update process, thereby enhancing performance. Wen et al. [7] applied a reinforcement learning-enhanced genetic algorithm to address multi-constraint scheduling in aircraft assembly, achieving notable results. Building on this foundation, the proposed approach proposes a reinforcement learning-based PSO algorithm, incorporating a Q-learning-driven parameter update mechanism. The algorithm dynamically adjusts the inertia weight and learning factors during iterations, promoting broader global exploration in the early stages and accelerating convergence in the later stages. By integrating Q-learning into the PSO framework, the proposed method aims to significantly improve the algorithm's performance in solving complex scheduling problems.

4.3.1 Markov Decision Model

The aircraft pulsating assembly line scheduling problem involves complex decision-making regarding AO precedence relationships, as well as the effects of resource and spatial constraints. These problems are characterized by temporality and randomness, making them well-suited to be modeled as a Markov Decision Process (MDP) for optimizing scheduling strategies. In this approach, the algorithm defines the state of the assembly line based on the current particle's completion time and categorizes it into five discrete states. It then designs six possible actions, selecting different actions for each state during the iteration process. This framework allows the algorithm to effectively manage uncertainties inherent in the assembly process, enhancing the adaptability and robustness of the scheduling plan. As a result, the overall production efficiency of the aircraft pulsating assembly line is significantly improved.

4.3.2 State Space

After modeling the aircraft pulsating assembly line scheduling problem as a Markov Decision Process (MDP), the algorithm takes the decoded completion time as the state variable. This completion time is divided into five discrete states for decision optimization. The algorithm obtains the completion time of 100 particle groups and calculates the average time $\overline{T}$. Depending on the scale size, an appropriate Δ value is chosen as the offset. The specific design is shown in Table 3:

Table 3: The set of state space

State	Partition interval of makespan
S_1	$\left[\overline{T}-5\Delta, \overline{T}-3\Delta\right]$
S_2	$\left[\overline{T}-3\Delta, \overline{T}-\Delta\right]$
S_3	$\left[\overline{T}-\Delta, \overline{T}+\Delta\right]$
S_4	$\left[\overline{T}+\Delta, \overline{T}+3\Delta\right]$
S_5	$\left[\overline{T}+3\Delta, \overline{T}+5\Delta\right]$

4.3.3 Action Space

After modeling the aircraft pulsating assembly line scheduling problem as a Markov Decision Process (MDP), the algorithm needs to design the action space, which will influence the scheduling strategy and optimize the final completion time state. This approach combines the optimization idea of reinforcement learning and defines the following six actions. Each action affects the AO sequence and guides the completion time to transition to different states. The specific design is shown in Table 4.

Table 4: The set of action space

Action	Parameter tuning of the PSO algorithm
A_1 (Global exploration)	$\omega = 1, c_1 = 2.5, c_2 = 0.5$
A_2 (Adjustment of inertia weight)	$\omega = rand$
A_3 (Local exploration)	$\omega = 0.8, c_1 = 2.0, c_2 = 1.0$
A_4 (Slow convergence)	$\omega = 0.6, c_1 = 1.0, c_2 = 2.0$
A_5 (Adjustment of learning factors)	$c_1 = rand, c_2 = rand$
A_6 (Fast convergence)	$\omega = 0.4, c_1 = 0.5, c_2 = 2.5$

4.3.4 Strategy Update Mechanism

After executing an action, the QL model will adjust the PSO parameters. The algorithm obtains the next generation of particles through the iterative update mechanism of PSO, calculates the completion time of each particle, and computes the corresponding reward based on the optimization objective. Here, T_f represents the completion time of the current particle, and T_{t-1} represents the average completion time of the previous generation's particles. If the value of T_f is less than T_{t-1}, the algorithm will receive a positive reward, indicating that the solution quality has improved. Conversely, if T_f is greater than T_{t-1}, the algorithm will receive a negative reward.

$$R_t\left(s_t, a_t\right) = -\frac{T_f - \overline{T}_{t-1}}{\overline{T}_{t-1}} \tag{15}$$

To achieve this optimization process, the QL model uses the basic Bellman iteration formula for updates. Using this formula, the current Q-value is updated based on the current reward obtained by the particle and the maximum Q-value in the next state, performing iterations in the QL model. Here, $\alpha = 0.2$, $\gamma = 0.8$.

$$Q^i\left(s_t, a_t\right) = Q^i\left(s_t, a_t\right) + \alpha\left[R^i\left(s_t, a_t\right) + \gamma \max_{a+1} Q^i\left(s_{t+1}, a_{t+1}\right) - Q^i\left(s_t, a_t\right)\right] \tag{16}$$

4.3.5 QLPSO Algorithmic Strategies and Processes

This study proposes the QLPSO algorithm, as illustrated in Fig. 5. The particle swarm parameters are initialized, and case data is read to generate parameters such as population size and the number of iterations. Each particle is encoded using a three-layer reverse order encoding scheme to represent feasible assembly sequences. Then, the particle positions and velocities are updated based on the standard Particle Swarm Optimization update rules. The AO sequences and their corresponding time lists are obtained through time decoding. The individual and global best positions are updated based on the final completion time. To avoid premature convergence, the population is reconstructed during the search process, retaining the top 60% of the population, while the remaining 40% is generated by new individuals for the next generation. Additionally, a Q-learning algorithm is employed to guide the dynamic adjustment of PSO parameters. The completion time of the current particle is used to obtain the corresponding state, update the Q-table, and apply a strategy to select actions that update the particle swarm parameters, continuing to the next generation. This adaptive mechanism balances global exploration and local exploitation, resulting in faster and more efficient scheduling optimization.

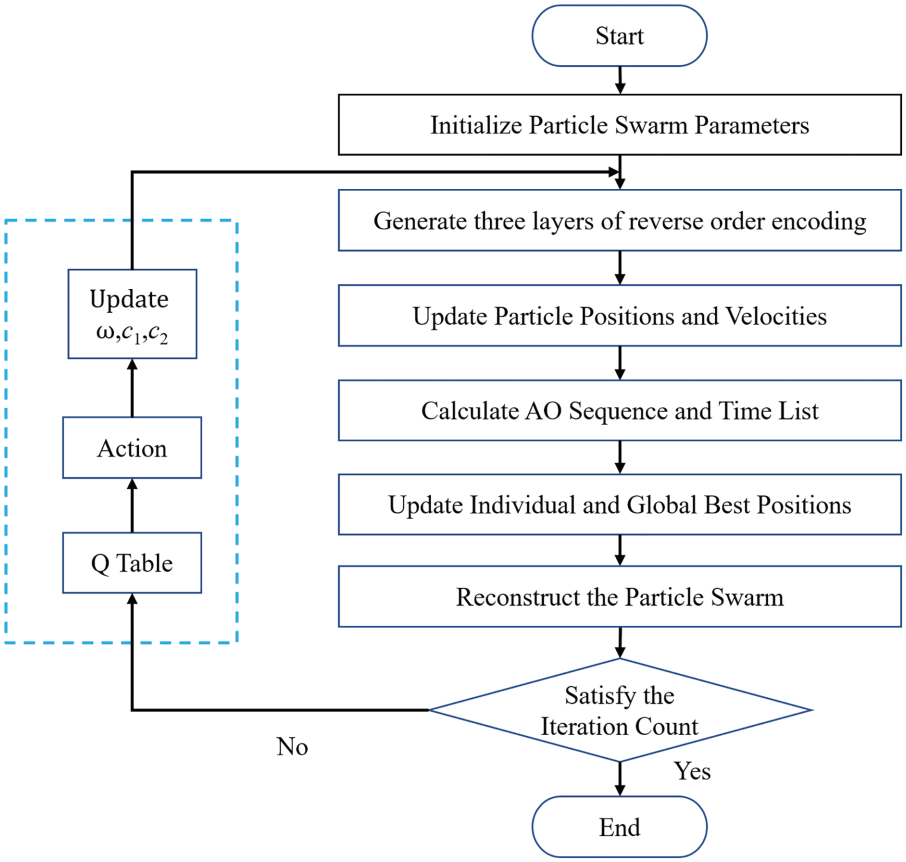


Figure 5: Encoding design of the aircraft pulsating assembly line scheduling problem

5 Experimental Results and Comparisons

5.1 Instance Validation

This article focuses on the aircraft pulsating assembly line of a large commercial aircraft as the research object. A total of 32 AO are designed to evaluate the feasibility and effectiveness of the proposed algorithm.

The resource requirements for components, task durations, operator A, operator B, tool A, and tool B are randomly generated. As the task set is based on a small-scale case, only two segments of the assembly line are selected for the experiment. The task names are referenced from Citation [7]. The available resources include 12 operator A personnel, 14 operator B personnel, 9 skilled operators, and a maximum of 10 units each for tool A and tool B. Task precedence relationships are defined using data files from the Resource-Constrained Project Scheduling Problem (RCPSP) benchmark [43]. Detailed task and resource information is presented in Table 5.

Table 5: Detailed information table of 32 AO

No.	AO	Segment	Duration	Operator A	Operator B	Tool A	Tool B	Subsequent
1	Pose adjustment	1	4	1	2	1	1	2, 3, 4, 5, 6
2	Fixed	1	8	3	4	2	1	12, 16, 21
3	Second mating	1	9	4	4	3	3	8, 13
4	Interior floor	2	10	5	5	4	3	7
5	Wire rigging	2	6	1	3	0	2	6
6	Intermediate antenna	2	8	1	6	1	4	14, 22
7	Low frequency antenna	1	10	6	3	2	5	9
8	Navigation antenna	1	8	3	2	2	2	12, 17
9	Emergency antenna	1	10	4	3	3	2	10
10	Satellite antenna	1	9	2	5	1	5	11
11	Main landing gear	2	3	1	1	0	1	16, 17, 22
12	Front landing gear	1	4	1	2	0	0	13
13	Data antenna	2	10	4	5	2	2	15
14	Traffic antenna	1	9	2	3	1	3	17
15	Height antenna	1	2	0	2	0	2	18, 22
16	Blind drop antenna	2	5	1	3	1	1	20
17	Main landing gear accessories	2	1	0	1	0	2	31
18	Front landing gear accessories	1	10	2	6	2	2	19, 22
19	Antenna sealing gluing	2	1	1	0	1	0	23
20	Igniting	2	7	2	4	1	2	25
21	Control valve	2	5	1	2	1	1	22
22	Power generation system	1	3	2	1	2	2	26, 28
23	Oil system	1	8	4	2	2	2	24
24	Engine gas pipe	2	2	1	1	1	1	27, 28
25	Degaussing equipment	1	10	4	3	2	3	29
26	Ventilation	2	1	0	1	0	1	30
27	Fuel system	1	4	1	2	1	1	28
28	Oil tank	1	2	1	1	1	1	32
29	Engine cable	2	5	1	3	1	2	32
30	Fuel pressure system	2	5	2	2	1	1	32
31	Quarter bend	1	3	0	1	0	2	32
32	Transfer	2	0	0	0	0	0	0

The algorithm programming environment is Visual Studio, with the code written in C++. The running environment consists of an Intel i7-12700F 2.10 GHz CPU and 32 GB of memory. Since the configuration of parameters significantly affects the performance of the QLPSO algorithm, this approach first determines the parameter selection range, as shown in Table 6. Since the algorithm proposed uses the QL model to modify the inertia weight ω and learning factor c_1, c_2 in the PSO algorithm, these parameters are not considered in the parameter settings for now.

Table 6: The setting of algorithm parameters

No.	Parameter						
	Population size	**Iteration**	P_{max}	V_{max}	α	γ	ε
1	100	100	10	3	0.2	0.8	0.2
2	100	200	10	3	0.2	0.8	0.2
3	100	300	10	3	0.2	0.8	0.2
4	100	100	2	2	0.4	0.9	0.8
5	100	200	2	2	0.4	0.9	0.8
6	100	300	2	2	0.4	0.9	0.8
7	200	100	10	3	0.2	0.8	0.2
8	200	200	10	3	0.2	0.8	0.2
9	200	300	10	3	0.2	0.8	0.2
10	200	100	2	2	0.4	0.9	0.8
11	200	200	2	2	0.4	0.9	0.8
12	200	300	2	2	0.4	0.9	0.8

Through experimental verification, the parameters selected are a population size of 200, 200 iterations, a maximum particle position (P_{max}) value of 10, a minimum (P_{min}) value of 0, a maximum particle velocity value (V_{max}) of 3, and a minimum value (V_{min}) of −3. In the QL model, the learning rate (α) is 0.2, the discount factor (γ) is 0.8, and the exploration probability (ε) is 0.2.

The QLPSO algorithm is employed in this study to solve the scheduling problem using the parameters described above. The results are illustrated in Fig. 6. The horizontal axis indicates the assembly time, while the vertical axis shows the allocation of skilled operators. Blue bars represent AO for Segment 1, and red bars represent AO for Segment 2. A higher vertical position corresponds to a greater number of operators assigned to a task. As shown in Fig. 6, the number of skilled operators allocated at any point does not exceed the total available. Figs. 7–9 further display the detailed allocation of skilled operators, as well as the individual schedules for Operator A and Operator B across different time intervals. These results demonstrate that operator assignments remain within capacity limits, confirming the feasibility and optimality of the solution under the defined constraints.

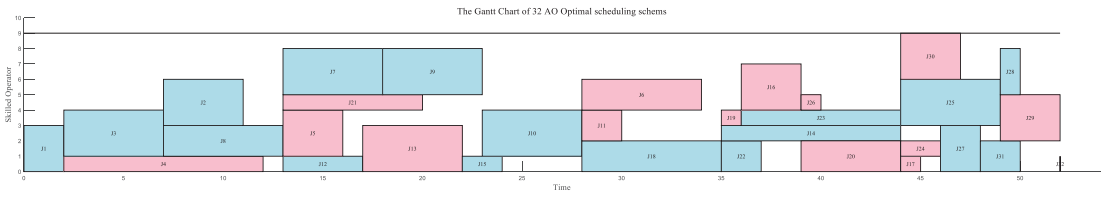


Figure 6: The Gantt chart of 32 AO optimal scheduling schemes for scheduling problem

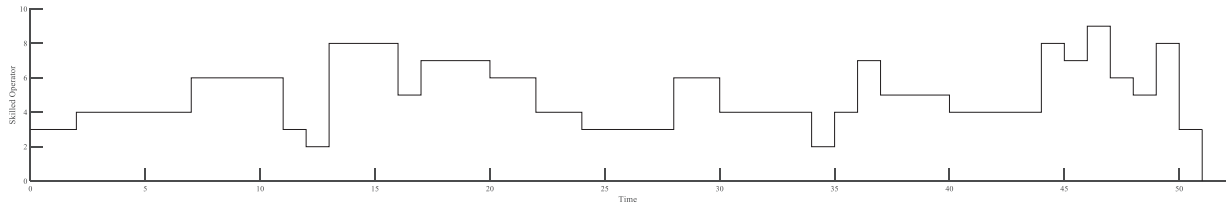


Figure 7: Ladder diagram of skilled operator resource allocation

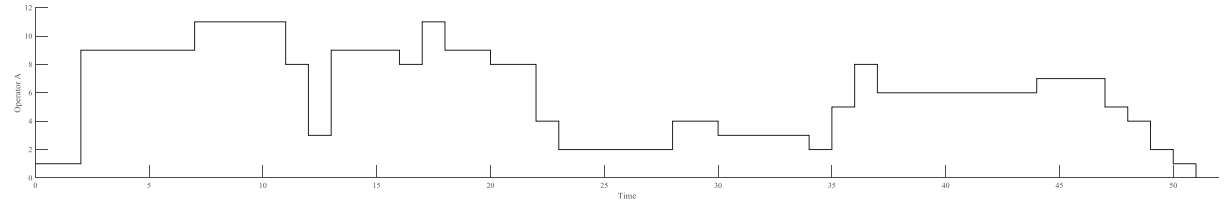


Figure 8: Ladder diagram of skilled Operator A resource allocation

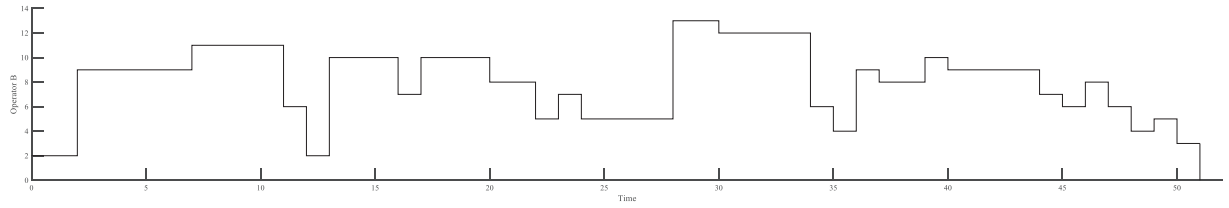


Figure 9: Ladder diagram of skilled Operator B resource allocation

5.2 Comparison Algorithm Implementation

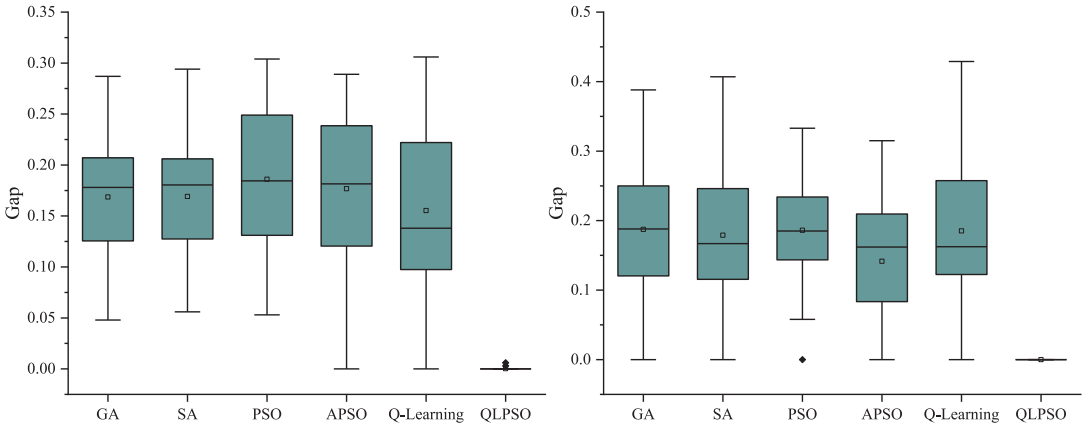
This study first reviewed relevant literature and then designed test cases of varying sizes: small-scale (20–30 AO), medium-scale (50–80 AO), large-scale (100–300 AO), and ultra-large-scale (600 AO). Six algorithms were evaluated: Genetic Algorithm (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO), Adaptive Particle Swarm Optimization (APSO), Q-learning, and the Q-Learning improved Particle Swarm Optimization algorithm (QLPSO) proposed in this article. For small and medium-scale instances, 100 test runs were conducted for each algorithm. For large-scale and ultra-large-scale cases, 60 and 20 runs were performed, respectively. In this context, Gap represents the difference between each algorithm's result and its optimal value. Here, T_2 denotes the optimal value obtained after the algorithmic computation.

$$Gap = \frac{T_1 - T_2}{T_2} \tag{17}$$

The average Gap for each scale is calculated, as the test cases vary across different scales. For example, the small-scale case involves 20 AO and 30 AO. Therefore, a box plot of Gap is drawn to more intuitively represent the algorithm's performance. The optimal and average results for each set of test cases are summarized in the table below. In each case, the optimal result is highlighted in bold.

The experimental results reveal that the average Gap values obtained by GA, SA, and PSO are relatively high, indicating substantial deviations from the optimal solutions. Although APSO (Min) and Q-learning (Avg) achieve certain improvements, with average Gap values of 14.1% and 15.5%, their performance remains inconsistent across different test cases. By contrast, the proposed QLPSO algorithm demonstrates superior performance, frequently attaining the optimal solution in all 40 test cases. This advantage is further corroborated by the boxplot, Fig. 10 analysis, in which QLPSO exhibits extremely low variance, thereby

confirming its high accuracy and remarkable stability. Overall, these findings suggest that QLPSO effectively integrates the strengths of PSO and Q-learning, providing a dynamic and adaptive approach that enhances both convergence speed and solution quality. A detailed comparison of the results is provided in Table 7.



(a) Box plot of average gap of the Avg value

(b) Box plot of average gap of the Min value

Figure 10: Box plot of the gap value for small-scale (20–30) AO instances

Table 7: Comparison of results for small-scale (20–30) AO instances

No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
20-1	43.95	41	44.56	41	43.55	38	43.36	38	43.48	40	**35.51**	**32**
Gap	0.238	0.281	0.255	0.281	0.226	0.188	0.221	0.188	0.224	0.250	**0.000**	**0.000**
20-2	55.79	49	55.37	49	54.78	48	54.85	49	53.07	49	**46.77**	**42**
Gap	0.193	0.167	0.184	0.167	0.171	0.143	0.173	0.167	0.135	0.167	**0.000**	**0.000**
20-3	50.58	44	51.28	45	50.88	44	49.85	41	49.17	45	**41.87**	**36**
Gap	0.208	0.222	0.225	0.250	0.215	0.222	0.191	0.139	0.174	0.250	**0.000**	**0.000**
20-4	43.73	38	43.07	**33**	43.21	39	42.77	**33**	41.34	38	**36.38**	**33**
Gap	0.202	0.152	0.184	**0.000**	0.188	0.182	0.176	**0.000**	0.136	0.152	**0.000**	**0.000**
20-5	45.82	38	45.76	38	46.37	38	45.37	39	44.15	38	**36.61**	**31**
Gap	0.252	0.226	0.250	0.226	0.267	0.226	0.239	0.258	0.206	0.226	**0.000**	**0.000**
20-6	46.83	41	46.98	41	46.56	41	46.78	41	45.11	39	**39.58**	**34**
Gap	0.183	0.206	0.187	0.206	0.176	0.206	0.182	0.206	0.140	0.147	**0.000**	**0.000**
20-7	46.53	40	46.27	40	46.63	42	45.89	**33**	43.93	41	**38.38**	**33**
Gap	0.212	0.212	0.206	0.212	0.215	0.273	0.196	**0.000**	0.145	0.242	**0.000**	**0.000**
20-8	42.1	37	42.33	38	42.12	34	41	34	41.38	35	**32.71**	**27**
Gap	0.287	0.370	0.294	0.407	0.288	0.259	0.253	0.259	0.265	0.296	**0.000**	**0.000**
20-9	52.96	47	52.97	47	53.17	47	52.87	46	51.49	45	**45.43**	**41**
Gap	0.166	0.146	0.166	0.146	0.170	0.146	0.164	0.122	0.133	0.098	**0.000**	**0.000**
20-10	49.45	43	49.53	43	49.97	**38**	49.43	45	48.36	45	**41.98**	**38**
Gap	0.178	0.132	0.180	0.132	0.190	**0.000**	0.177	0.184	0.152	0.184	**0.000**	**0.000**
20-11	62.21	57	62.75	57	62.45	55	62.95	57	60.31	54	**53.31**	**48**
Gap	0.167	0.188	0.177	0.188	0.171	0.146	0.181	0.188	0.131	0.125	**0.000**	**0.000**
20-12	62.16	54	61.77	55	62.01	56	62.28	57	60.11	56	**53.61**	**49**
Gap	0.159	0.102	0.152	0.122	0.157	0.143	0.162	0.163	0.121	0.143	**0.000**	**0.000**
20-13	54.92	47	54.87	49	54.13	48	54.46	45	52.25	46	**43.45**	**37**
Gap	0.264	0.270	0.263	0.324	0.246	0.297	0.253	0.216	0.203	0.243	**0.000**	**0.000**
20-14	60.25	55	60.14	53	61.36	55	61.13	57	59.17	54	**51.56**	**47**
Gap	0.169	0.170	0.166	0.128	0.190	0.170	0.186	0.213	0.148	0.149	**0.000**	**0.000**
20-15	62.11	57	62.2	56	61.96	57	61.1	56	59.95	55	**53.52**	**48**

(Continued)

Table 7 (continued)

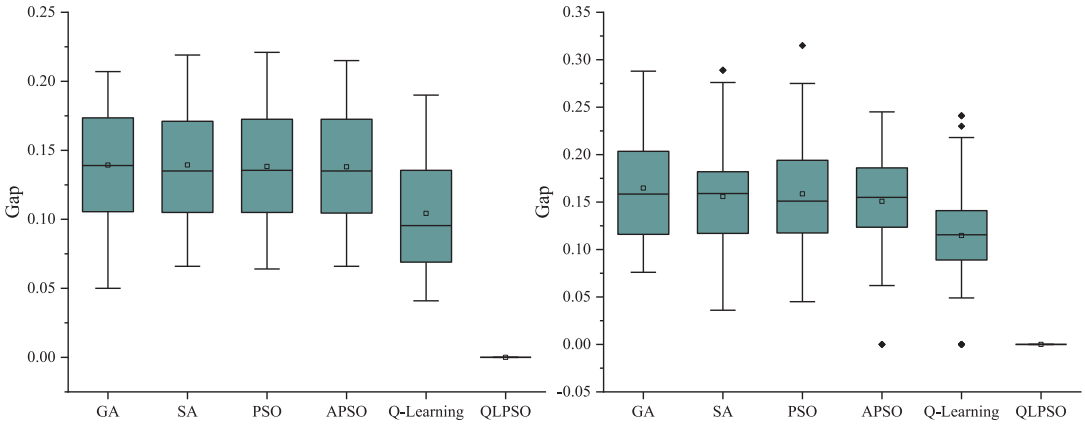
No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Gap	0.161	0.188	0.162	0.167	0.158	0.188	0.142	0.167	0.120	0.146	**0.000**	**0.000**
20-16	54.77	46	55.44	**44**	55.44	52	54.67	**44**	53.9	**44**	**48.24**	**44**
Gap	0.135	0.045	0.149	**0.000**	0.149	0.182	0.133	**0.000**	0.117	**0.000**	**0.000**	**0.000**
20-17	62.95	57	63.05	57	62.47	56	62.49	57	60.17	56	**53.18**	**47**
Gap	0.184	0.213	0.186	0.213	0.175	0.191	0.175	0.213	0.131	0.191	**0.000**	**0.000**
20-18	69.11	60	68.26	60	69.33	64	69.42	64	64.79	59	**58.69**	**52**
Gap	0.178	0.154	0.163	0.154	0.181	0.231	0.183	0.231	0.104	0.135	**0.000**	**0.000**
20-19	44.31	39	44.93	41	44.16	39	43.98	**33**	43.19	39	**38.05**	**33**
Gap	0.165	0.182	0.181	0.242	0.161	0.182	0.156	**0.000**	0.135	0.182	**0.000**	**0.000**
20-20	51.38	46	52.21	44	51.2	47	51.17	45	50.29	44	**42.03**	**38**
Gap	0.222	0.211	0.242	0.158	0.218	0.237	0.217	0.184	0.197	0.158	**0.000**	**0.000**
30-1	76.78	70	76.77	70	78.74	67	78.58	65	76.38	56	**63.68**	**56**
Gap	0.206	0.250	0.206	0.250	0.236	0.196	0.234	0.161	0.199	0.000	**0.000**	**0.000**
30-2	94.17	87	94.04	86	98.45	86	98.77	86	96.76	86	**79.3**	**68**
Gap	0.188	0.279	0.186	0.265	0.241	0.265	0.246	0.265	0.220	0.265	**0.000**	**0.000**
30-3	73.37	65	72.7	67	77.56	62	76.53	62	77.69	67	**59.49**	**52**
Gap	0.233	0.250	0.222	0.288	0.304	0.192	0.286	0.192	0.306	0.288	**0.000**	**0.000**
30-4	89.11	79	89.2	80	92.64	78	93.37	77	90.95	85	**74.01**	**66**
Gap	0.204	0.197	0.205	0.212	0.252	0.182	0.262	0.167	0.229	0.288	**0.000**	**0.000**
30-5	76.7	72	76.33	**54**	82.7	68	81.9	**54**	81.83	74	**65.32**	**54**
Gap	0.174	0.333	0.169	**0.000**	0.266	0.259	0.254	**0.000**	0.253	0.370	**0.000**	**0.000**
30-6	83.46	76	83.48	73	86.59	72	84.65	74	83.99	77	**68.39**	**60**
Gap	0.220	0.267	0.221	0.217	0.266	0.200	0.238	0.233	0.228	0.283	**0.000**	**0.000**
30-7	76.79	72	76.14	71	80.81	72	81.25	71	77.81	71	**63.03**	**54**
Gap	0.218	0.333	0.208	0.315	0.282	0.333	0.289	0.315	0.234	0.315	**0.000**	**0.000**
30-8	75.15	68	76.37	67	83.48	71	80.88	**55**	83.1	75	**65.75**	**55**
Gap	0.143	0.236	0.162	0.218	0.270	0.291	0.230	**0.000**	0.264	0.364	**0.000**	**0.000**
30-9	71.17	65	71.46	63	76.65	65	75.87	56	74.84	65	**59.97**	**49**
Gap	0.187	0.327	0.192	0.286	0.278	0.327	0.265	0.143	0.248	0.327	**0.000**	**0.000**
30-10	73.4	68	72.66	67	77.44	63	77.38	61	77.34	70	**61.32**	**49**
Gap	0.197	0.388	0.185	0.367	0.263	0.286	0.262	0.245	0.261	0.429	**0.000**	**0.000**
30-11	107.25	**90**	107.01	100	107.88	103	107.92	101	**98.49**	**90**	98.75	**90**
Gap	0.088	**0.000**	0.086	0.111	0.095	0.144	0.095	0.122	**0.00**	**0.000**	0.003	**0.000**
30-12	120.56	113	120.53	113	120.33	109	120.15	105	118.82	112	**112.75**	**103**
Gap	0.069	0.097	0.069	0.097	0.067	0.058	0.066	0.019	0.054	0.087	**0.000**	**0.000**
30-13	95.67	**84**	96.31	88	96.13	90	95.96	89	95.5	91	**90.04**	**84**
Gap	0.063	**0.000**	0.070	0.048	0.068	0.071	0.065	0.060	0.061	0.083	**0.000**	**0.000**
30-14	109.79	102	109.43	102	108.85	103	109.49	100	107.94	100	**99.67**	**89**
Gap	0.102	0.146	0.098	0.146	0.092	0.157	0.099	0.124	0.083	0.124	**0.000**	**0.000**
30-15	114.82	107	114.83	108	114.84	106	114.21	108	114.24	109	**106.84**	**99**
Gap	0.075	0.081	0.075	0.091	0.075	0.071	0.069	0.091	0.069	0.101	**0.000**	**0.000**
30-16	111.65	103	109.93	102	111.25	104	111.37	102	109.84	104	**101.59**	**92**
Gap	0.099	0.120	0.082	0.109	0.095	0.130	0.096	0.109	0.081	0.130	**0.000**	**0.000**
30-17	110.73	100	109.72	97	110.18	100	109.96	100	108.03	102	**99.22**	**91**
Gap	0.116	0.099	0.106	0.066	0.110	0.099	0.108	0.099	0.089	0.121	**0.000**	**0.000**
30-18	104.65	98	105.43	98	105.12	99	104.65	99	104.61	99	**99.82**	**92**
Gap	0.048	0.065	0.056	0.065	0.053	0.076	0.048	0.076	0.048	0.076	**0.000**	**0.000**
30-19	112.78	102	112.34	106	113.58	107	112.22	104	111.32	107	**102.07**	**91**
Gap	0.105	0.121	0.101	0.165	0.113	0.176	0.099	0.143	0.091	0.176	**0.000**	**0.000**
30-20	98.21	89	98.79	93	99.5	93	**90.21**	**83**	97.38	91	90.82	**83**
Gap	0.088	0.072	0.095	0.120	0.102	0.120	**0.000**	**0.000**	0.079	0.096	0.006	**0.000**

(Continued)

Table 7 (continued)

No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
average Gap	16.9%	18.7%	16.9%	17.9%	18.6%	18.6%	17.7%	14.1%	15.5%	18.5%	**0.2%**	**0**

In the medium-scale test cases, the QLPSO algorithm continues to deliver the best overall performance. Traditional algorithms such as GA, SA, PSO, and APSO exhibit relatively high and widely distributed average Gap values, indicating significant performance variations across different instances and a noticeable deviation from the optimal solution. In contrast, the Q-learning algorithm demonstrates relatively better results, with average Gap values of 10.4% and 11.4%, respectively. Notably, QLPSO outperforms all other algorithms, consistently achieving optimal or near-optimal solutions across the given test cases. The box plot for the average Gap of Min value (Fig. 11b) shows a significantly tighter distribution, with values much closer to 0, further confirming the superior performance of QLPSO. This suggests that the algorithm is capable of finding the best possible solution in most cases, highlighting its stability and effectiveness. A detailed comparison of the results is presented in Table 8.



(a) Box plot of average gap of the Avg value

(b) Box plot of average gap of the Min value

Figure 11: Box plot of the gap value for medium-scale (50–80) AO instances

Table 8: Comparison of results for medium-scale (50–80) AO instances

No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
50-1	198.96	187	198.19	187	198.25	188	197.82	188	193.44	183	**182.97**	**168**
Gap	0.087	0.113	0.083	0.113	0.084	0.119	0.081	0.119	0.057	0.089	**0.000**	**0.000**
50-2	169.21	156	168.4	156	168.27	158	169.08	157	164.08	**145**	**155.95**	**145**
Gap	0.085	0.076	0.080	0.076	0.079	0.090	0.084	0.083	0.052	**0.000**	**0.000**	**0.000**
50-3	187.29	176	187.14	177	187.75	179	187.41	172	184.07	170	**174.01**	**162**
Gap	0.076	0.086	0.075	0.093	0.079	0.105	0.077	0.062	0.058	0.049	**0.000**	**0.000**
50-4	210.05	198	211.56	193	210.93	197	210.61	198	204.74	192	**192.08**	**177**
Gap	0.094	0.119	0.101	0.090	0.098	0.113	0.096	0.119	0.066	0.085	**0.000**	**0.000**
50-5	157.73	149	157.07	148	157.73	148	157.85	150	152.62	145	**144.66**	**133**
Gap	0.090	0.120	0.086	0.113	0.090	0.113	0.091	0.128	0.055	0.090	**0.000**	**0.000**
50-6	156.88	145	157.31	142	157.34	149	157.75	147	151.5	143	**139.69**	**130**

(Continued)

Table 8 (continued)

No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Gap	0.123	0.115	0.126	0.092	0.126	0.146	0.129	0.131	0.085	0.100	**0.000**	**0.000**
50-7	171.58	157	171.62	159	170.99	160	171.37	162	166.81	156	**156.74**	**145**
Gap	0.095	0.083	0.095	0.097	0.091	0.103	0.093	0.117	0.064	0.076	**0.000**	**0.000**
50-8	151.75	146	154.17	145	153.81	145	154.06	148	150.54	144	**144.57**	**134**
Gap	0.050	0.090	0.066	0.082	0.064	0.082	0.066	0.104	0.041	0.075	**0.000**	**0.000**
50-9	171.69	160	171.89	161	171.77	160	171.49	159	164.96	158	**153.63**	**137**
Gap	0.118	0.168	0.119	0.175	0.118	0.168	0.116	0.161	0.074	0.153	**0.000**	**0.000**
50-10	150.57	136	150.57	142	150.71	142	150.67	142	144.51	136	**136.72**	**124**
Gap	0.101	0.097	0.101	0.145	0.102	0.145	0.102	0.145	0.057	0.097	**0.000**	**0.000**
50-11	188.49	174	187.56	176	188.06	177	187.61	179	182.42	173	**173.21**	**157**
Gap	0.088	0.108	0.083	0.121	0.086	0.127	0.083	0.140	0.053	0.102	**0.000**	**0.000**
50-12	185.37	172	185.91	175	185.52	161	184.87	166	177.88	168	**165.39**	**154**
Gap	0.121	0.117	0.124	0.136	0.122	0.045	0.118	0.078	0.076	0.091	**0.000**	**0.000**
50-13	139.26	129	140.04	130	139.06	125	139.89	130	134.44	**112**	**124.93**	**112**
Gap	0.115	0.152	0.121	0.161	0.113	0.116	0.120	0.161	0.076	**0.000**	**0.000**	**0.000**
50-14	184.32	175	184.13	164	184.5	171	183.86	173	178	164	**166.09**	**151**
Gap	0.110	0.159	0.109	0.086	0.111	0.132	0.107	0.146	0.072	0.086	**0.000**	**0.000**
50-15	195.62	185	195.69	174	195.55	181	196.09	**168**	191.29	183	**179.38**	**168**
Gap	0.091	0.101	0.091	0.036	0.090	0.077	0.093	**0.000**	0.066	0.089	**0.000**	**0.000**
50-16	158.86	149	158.58	146	158.43	147	158.12	146	151.72	142	**140.02**	**129**
Gap	0.135	0.155	0.133	0.132	0.131	0.140	0.129	0.132	0.084	0.101	**0.000**	**0.000**
50-17	172.33	161	171.82	162	170.61	156	171.41	160	165.33	157	**153**	**139**
Gap	0.126	0.158	0.123	0.165	0.115	0.122	0.120	0.151	0.081	0.129	**0.000**	**0.000**
50-18	168.93	159	168.99	156	168.37	158	169.28	157	163.91	155	**151.91**	**138**
Gap	0.112	0.152	0.112	0.130	0.108	0.145	0.114	0.138	0.079	0.123	**0.000**	**0.000**
50-19	176.26	159	176.61	167	175.53	158	176.15	165	171.43	161	**157.75**	**144**
Gap	0.117	0.104	0.120	0.160	0.113	0.097	0.117	0.146	0.087	0.118	**0.000**	**0.000**
50-20	153.02	143	153.36	142	152.18	163	152.61	**124**	146.04	136	**135.31**	**124**
Gap	0.131	0.153	0.133	0.145	0.125	0.315	0.128	**0.000**	0.079	0.097	**0.000**	**0.000**
80-1	151.5	143	151.64	140	150.56	140	151.9	140	146.51	136	**129.65**	**119**
Gap	0.169	0.202	0.170	0.176	0.161	0.176	0.172	0.176	0.130	0.143	**0.000**	**0.000**
80-2	111.62	103	111.2	100	110.72	98	111.54	104	107.78	100	**94.22**	**87**
Gap	0.185	0.184	0.180	0.149	0.175	0.126	0.184	0.195	0.144	0.149	**0.000**	**0.000**
80-3	144.18	135	144.34	135	148.18	136	147.46	137	144.4	134	**121.36**	**110**
Gap	0.188	0.227	0.189	0.227	0.221	0.236	0.215	0.245	0.190	0.218	**0.000**	**0.000**
80-4	109.54	103	109.22	97	110	102	109.3	95	105.76	**80**	**91.28**	**80**
Gap	0.200	0.288	0.197	0.213	0.205	0.275	0.197	0.188	0.159	**0.000**	**0.000**	**0.000**
80-5	116.44	109	116.8	111	116.3	106	116.78	108	113.29	107	**98.8**	**87**
Gap	0.179	0.253	0.182	0.276	0.177	0.218	0.182	0.241	0.147	0.230	**0.000**	**0.000**
80-6	116.82	104	117.92	107	117.24	103	116.9	103	114.39	103	**97.21**	**83**
Gap	0.202	0.253	0.213	0.289	0.206	0.241	0.203	0.241	0.177	0.241	**0.000**	**0.000**
80-7	110.84	100	110.12	101	109.8	104	108.88	99	106.93	100	**92.07**	**85**
Gap	0.204	0.176	0.196	0.188	0.193	0.224	0.183	0.165	0.161	0.176	**0.000**	**0.000**
80-8	111.32	104	112.42	107	110.28	99	109.6	98	108.62	103	**92.23**	**83**
Gap	0.207	0.253	0.219	0.289	0.196	0.193	0.188	0.181	0.178	0.241	**0.000**	**0.000**
80-9	112.98	106	112.62	103	113.18	104	113.08	102	109.58	100	**96.35**	**88**
Gap	0.173	0.205	0.169	0.170	0.175	0.182	0.174	0.159	0.137	0.136	**0.000**	**0.000**
80-10	113.8	106	113.38	105	113.24	104	111.52	103	109.49	102	**95.07**	**87**
Gap	0.197	0.218	0.193	0.207	0.191	0.195	0.173	0.184	0.152	0.172	**0.000**	**0.000**
80-11	192.28	184	192.7	178	192.54	178	192.26	181	186.77	**150**	**165.18**	**150**
Gap	0.164	0.227	0.167	0.187	0.166	0.187	0.164	0.207	0.131	**0.000**	**0.000**	**0.000**
80-12	179.78	171	179	164	179.2	167	180.06	173	173.46	157	**153.12**	**141**

(Continued)

Table 8 (continued)

No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Gap	0.174	0.213	0.169	0.163	0.170	0.184	0.176	0.227	0.133	0.113	**0.000**	**0.000**
80-13	192.7	182	190.94	176	192.22	182	192.88	182	186.19	171	**165.17**	**152**
Gap	0.167	0.197	0.156	0.158	0.164	0.197	0.168	0.197	0.127	0.125	**0.000**	**0.000**
80-14	190.52	179	189.68	174	189.9	175	190.58	180	184.96	169	**163.8**	**150**
Gap	0.163	0.193	0.158	0.160	0.159	0.167	0.163	0.200	0.129	0.127	**0.000**	**0.000**
80-15	182.6	169	182.44	173	182.6	170	182.5	171	176.59	165	**159.8**	**147**
Gap	0.143	0.150	0.142	0.177	0.143	0.156	0.142	0.163	0.105	0.122	**0.000**	**0.000**
80-16	189.68	181	189.78	178	189.2	175	188.68	179	184.04	172	**165.32**	**152**
Gap	0.147	0.191	0.148	0.171	0.144	0.151	0.141	0.178	0.113	0.132	**0.000**	**0.000**
80-17	187.67	176	186.02	172	186.5	178	186.62	170	180.56	169	**163.57**	**153**
Gap	0.147	0.150	0.137	0.124	0.140	0.163	0.141	0.111	0.104	0.105	**0.000**	**0.000**
80-18	218.02	207	219.32	209	217.42	209	217.9	197	211.58	197	**186.54**	**173**
Gap	0.169	0.197	0.176	0.208	0.166	0.208	0.168	0.139	0.134	0.139	**0.000**	**0.000**
80-19	180.76	170	181.64	168	181.08	168	181.26	171	175.34	164	**156.59**	**146**
Gap	0.154	0.164	0.160	0.151	0.156	0.151	0.158	0.171	0.120	0.123	**0.000**	**0.000**
80-20	196.18	187	195.78	184	196.92	186	194.96	184	190.44	176	**167.05**	**153**
Gap	0.174	0.222	0.172	0.203	0.179	0.216	0.167	0.203	0.140	0.150	**0.000**	**0.000**
average Gap	13.9%	16.5%	13.9%	15.6%	13.8%	15.9%	13.8%	15.0%	10.4%	11.4%	**0**	**0**

In the large-scale test cases, the comparison of algorithm performance on large-scale instances reveals significant differences in their ability to reach the optimal solution. QLPSO outperforms all other algorithms, with an average Gap of only 0.05% and a minimum Gap of 0, indicating its strong robustness and consistent ability to find near-optimal solutions. The Q-learning algorithm also performs well, with an average Gap of 7.2% for the minimum values, while other algorithms show higher average Gaps, around 10%, with greater variation, indicating poorer stability and a lower likelihood of finding the optimal solution. These results suggest that QLPSO improves both the quality and consistency of the solutions, demonstrating the best solution quality and the strongest stability. Overall, QLPSO exhibits the optimal solution quality and stability in large-scale test cases, with Q-learning in second place, while other methods are relatively inferior. A detailed comparison of the results is presented in Table 9 and Fig. 12.

Table 9: Comparison of results for large-scale (100–300) AO instances

No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
100-1	259	248	256.15	240	256.97	241	259.37	248	249.3	240	**232.53**	**221**
Gap	0.114	0.122	0.102	0.086	0.105	0.090	0.115	0.122	0.072	0.086	**0.000**	**0.000**
100-2	293.5	284	297.8	287	292.9	283	292.57	278	285.87	277	**264.07**	**247**
Gap	0.111	0.150	0.128	0.162	0.109	0.146	0.108	0.126	0.083	0.121	**0.000**	**0.000**
100-3	217.65	208	216.9	205	217.6	208	216.6	203	209.5	201	**189.43**	**177**
Gap	0.149	0.175	0.145	0.158	0.149	0.175	0.143	0.147	0.106	0.136	**0.000**	**0.000**
100-4	212.95	197	212.15	205	212.13	199	211.2	202	205.93	199	**188.8**	**178**
Gap	0.128	0.107	0.124	0.152	0.124	0.118	0.119	0.135	0.091	0.118	**0.000**	**0.000**
100-5	210.05	203	211.35	204	211.93	203	211.87	188	204.45	**176**	**187.9**	**176**
Gap	0.118	0.153	0.125	0.159	0.128	0.153	0.128	0.068	0.088	**0.000**	**0.000**	**0.000**
100-6	315.65	307	313.25	303	313	291	313.27	301	303.2	287	**284.5**	**270**
Gap	0.109	0.137	0.101	0.122	0.100	0.078	0.101	0.115	0.066	0.063	**0.000**	**0.000**
100-7	331.3	315	332.85	322	332.67	314	335.23	319	324.7	310	**299.9**	**286**
Gap	0.105	0.101	0.110	0.126	0.109	0.098	0.118	0.115	0.083	0.084	**0.000**	**0.000**
100-8	233.2	224	233.1	222	232.17	218	233.97	220	225.67	212	**210.77**	**199**

(Continued)

Table 9 (continued)

No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
Gap	0.106	0.126	0.106	0.116	0.102	0.095	0.110	0.106	0.071	0.065	**0.000**	**0.000**
100-9	340.95	322	336.3	320	339.8	329	339.65	302	332.1	316	**314.2**	**302**
Gap	0.085	0.066	0.070	0.060	0.081	0.089	0.081	0.000	0.057	0.046	**0.000**	**0.000**
100-10	364.8	343	366.1	355	364.43	352	362.8	333	356.83	340	**336.1**	**325**
Gap	0.085	0.055	0.089	0.092	0.084	0.083	0.079	0.025	0.062	0.046	**0.000**	**0.000**
300-1	327.7	322	323.6	315	324.9	315	326.3	309	309.3	268	**274.58**	**254**
Gap	0.071	0.126	0.100	0.126	0.074	0.126	0.098	0.138	0.081	0.118	**0.000**	**0.000**
300-2	274.5	266	270.1	249	264.55	243	266.4	254	259.15	232	**222.4**	**217**
Gap	0.278	0.120	0.180	0.120	0.189	0.120	0.324	0.189	0.167	0.143	**0.000**	**0.000**
300-3	480	472	485.7	473	483.85	470	484.9	475	474.7	464	**447.1**	**438**
Gap	0.055	0.057	0.032	0.014	0.083	0.073	0.109	0.050	0.000	0.025	**0.000**	**0.000**
300-4	440.6	432	445.5	433	433.1	429	444.5	428	**406.56**	**399**	406.8	**399**
Gap	0.083	0.033	0.095	0.073	0.065	0.075	0.093	0.060	**0.000**	**0.000**	0.001	**0.000**
300-5	431.9	424	429.9	417	434.6	430	430.7	419	420	410	**391.1**	**382**
Gap	0.134	0.123	0.089	0.052	0.111	0.126	0.180	0.079	0.078	0.113	**0.000**	**0.000**
300-6	421.6	414	420.3	404	421.6	404	420.7	409	412.6	401	**384.9**	**372**
Gap	0.129	0.156	0.095	0.169	0.095	0.086	0.107	0.027	0.107	0.062	**0.000**	**0.000**
300-7	446.8	442	449.9	444	443.7	428	443.2	425	436.8	430	**406**	**386**
Gap	0.038	0.062	0.093	0.111	0.093	0.109	0.136	0.052	0.038	0.000	**0.000**	**0.000**
300-8	428.5	423	428.4	423	425.9	412	427.5	417	417.4	413	**386.4**	**378**
Gap	0.148	0.063	0.102	0.138	0.102	0.090	0.194	0.063	0.194	0.074	**0.000**	**0.000**
300-9	459.5	454	456.3	445	461.3	452	461.8	457	449.7	439	**424.2**	**417**
Gap	0.013	0.043	0.017	0.014	0.087	0.084	0.013	0.043	0.087	0.041	**0.000**	**0.000**
300-10	460.3	452	465.6	457	461.5	439	461.4	456	454.1	432	**427.5**	**420**
Gap	0.038	0.024	0.038	0.029	0.080	0.045	0.017	0.021	0.080	0.064	**0.000**	**0.000**
average Gap	10.4%	9.9%	9.7%	10.4%	10.3%	10.3%	11.8%	8.4%	8.0%	7.2%	**0.05%**	**0**

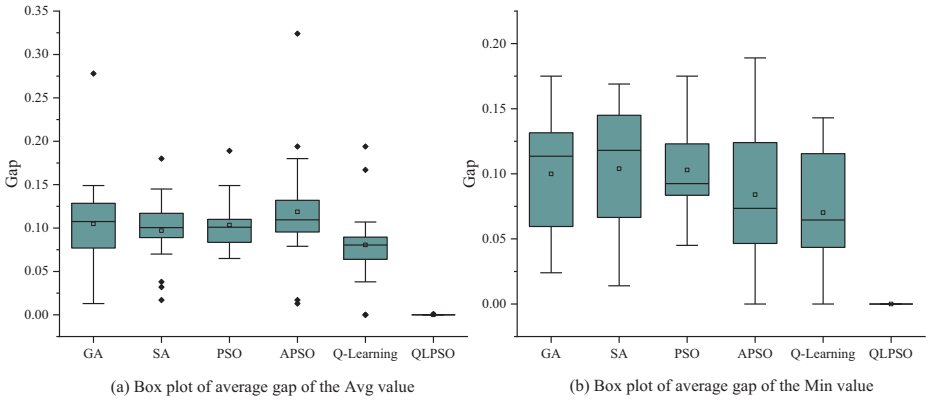
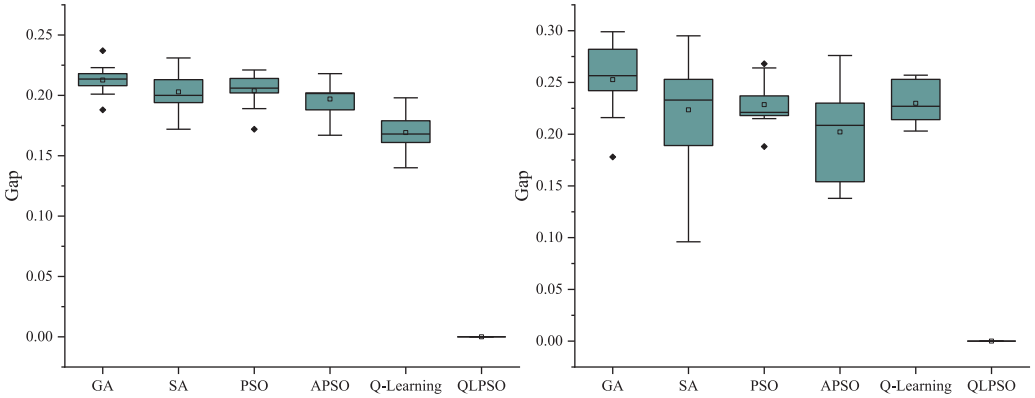


Figure 12: Box plot of the gap value for large-scale (100–300) AO instances

In the ultra-large-scale test cases, In the results of the 600 AO cases, QLPSO consistently outperforms all other algorithms, with the average Gap of Avg value (0%) and the average Gap of Min value (0%), indicating its high stability in finding near-optimal or optimal solutions. Compared to the optimal value, QLPSO's algorithm results are 25.2% better than those of the GA algorithm. Other algorithms exhibit significantly higher average Gaps (ranging from 16.9% to 25.2%) with a broader distribution, indicating that their solutions

deviate more from the optimal solution and have poorer consistency. The box plot, Fig. 13, further confirms that QLPSO demonstrates stable algorithm performance, while the distribution range of other algorithms is much wider, indicating that their solutions are less consistent and less effective in large-scale scenarios. A detailed comparison of the results is presented in Table 10.



(a) Box plot of average gap of the Avg value

(b) Box plot of average gap of the Min value

Figure 13: Box plot of the gap value for ultra-large-scale (600) AO instances

Table 10: Comparison of results for ultra-large-scale (600) AO instances

No.	GA		SA		PSO		APSO		Q-learning		QLPSO	
	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min	Avg	Min
600-1	688.83	656	678.6	638	681.2	638	690.4	644	658	644	**566.8**	**524**
Gap	0.215	0.252	0.197	0.218	0.202	0.218	0.218	0.229	0.161	0.229	**0.000**	**0.000**
600-2	716.67	690	699.2	596	715	690	696.2	624	701.6	684	**585.8**	**544**
Gap	0.223	0.268	0.194	0.096	0.221	0.268	0.188	0.147	0.198	0.257	**0.000**	**0.000**
600-3	729.2	698	726.18	710	732.8	710	724.8	698	712	706	**603.4**	**574**
Gap	0.208	0.216	0.203	0.237	0.214	0.237	0.201	0.216	0.180	0.230	**0.000**	**0.000**
600-4	714.4	650	697.4	656	707.6	656	717.4	650	695.2	664	**595**	**552**
Gap	0.201	0.178	0.172	0.188	0.189	0.188	0.206	0.178	0.168	0.203	**0.000**	**0.000**
600-5	729.8	706	719.2	666	724.2	708	715.2	646	698.4	678	**602**	**560**
Gap	0.212	0.261	0.195	0.189	0.203	0.264	0.188	0.154	0.160	0.211	**0.000**	**0.000**
600-6	713.8	684	714.6	676	703.8	668	701.2	626	684.8	674	**600.6**	**550**
Gap	0.188	0.244	0.190	0.229	0.172	0.215	0.167	0.138	0.140	0.225	**0.000**	**0.000**
600-7	708.4	694	714.4	674	700.4	658	700	664	687	678	**582.6**	**540**
Gap	0.216	0.285	0.226	0.248	0.202	0.219	0.202	0.230	0.179	0.256	**0.000**	**0.000**
600-8	712.4	678	709.4	684	707	666	703.2	656	683	666	**585**	**546**
Gap	0.218	0.242	0.213	0.253	0.209	0.220	0.202	0.201	0.168	0.220	**0.000**	**0.000**
600-9	703	678	702.8	676	705.8	644	695.8	666	677.8	654	**581.6**	**522**
Gap	0.209	0.299	0.208	0.295	0.214	0.234	0.196	0.276	0.165	0.253	**0.000**	**0.000**
600-10	711.8	682	707.8	682	696.9	650	691.2	666	674.8	646	**575.2**	**532**
Gap	0.237	0.282	0.231	0.282	0.212	0.222	0.202	0.252	0.173	0.214	**0.000**	**0.000**
average Gap	21.2%	25.2%	20.3%	22.4%	20.4%	22.8%	19.7%	20.2%	16.9%	22.9%	**0**	**0**

The experimental results across different test case scales consistently demonstrate the superior performance of the proposed QLPSO algorithm. In the small-scale test cases, QLPSO outperforms traditional algorithms such as GA, SA, PSO, and APSO, achieving the Avg and Min values, while exhibiting exceptional stability across all test cases. In the medium- and large-scale instances, QLPSO continues to deliver optimal or near-optimal solutions, with both the average Gap of the Avg value and the average Gap of the Min value lower than those of other algorithms, highlighting its robustness and consistent ability to find near-optimal solutions. The Q-learning algorithm also performs well but shows greater variation, with higher Gap values and lower consistency compared to QLPSO. In the ultra-large-scale test cases, involving 600 AO cases, QLPSO again demonstrates its exceptional stability and accuracy, achieving the best results with a Gap of 0% across all cases, 25.2% better than GA. The box plot analysis further emphasizes the superior consistency and performance of QLPSO, with a significantly tighter distribution compared to other algorithms. Overall, these results clearly indicate that QLPSO not only provides the best solution quality but also ensures the strongest stability across all scales, outperforming all other methods and offering a dynamic and adaptive approach that integrates the strengths of PSO and Q-learning.

6 Conclusions

This study presents a comprehensive study of the scheduling problem associated with aircraft pulsating assembly lines in the aircraft manufacturing industry. It systematically reviews relevant theories related to the resource-constrained project scheduling problem (RCPSP) and evaluates the current advancements in aircraft assembly line scheduling research. Building on this foundation, a scheduling model that integrates skilled operator allocation for the pulsating assembly line is proposed. To solve this model, a Q-Learning improved Particle Swarm Optimization algorithm is developed. The algorithm incorporates a Q-table and defines appropriate state and action spaces. The practical applicability and solution accuracy of the model are validated through real-world case studies. Finally, this paper outlines two key directions for future research in aircraft assembly line scheduling.

(1) The current model considers only the allocation of skilled operators and their impact on task durations. However, in real production environments, task durations are often subject to considerable uncertainty. Furthermore, operators differ in terms of types and skill levels, which further complicates the scheduling process. These factors introduce additional layers of complexity, posing significant challenges for future research on scheduling aircraft pulse assembly lines.

(2) Although the proposed QLPSO algorithm demonstrates strong performance in addressing the aircraft pulse assembly line scheduling problem, the application of reinforcement learning in this domain remains relatively limited. Furthermore, its implementation still requires further investigation. For example, the design of state and action spaces remains complex, and achieving an effective balance between exploration and exploitation continues to demand careful refinement and optimization. Future research could focus on developing more efficient algorithms or enhancing existing ones to improve both solution accuracy and computational efficiency.

In conclusion, this study provides both a theoretical foundation and a practical reference for optimizing scheduling in aircraft pulse assembly lines. It is hoped that future research will continue to build upon and further refine the findings presented herein.

Acknowledgement: Thanks to the professors for their valuable guidance and support throughout this research.

Funding Statement: This research was supported by the National Natural Science Foundation of China (Grant No. 52475543), Natural Science Foundation of Henan (Grant No. 252300421101), Henan Province University Science and

Technology Innovation Talent Support Plan (Grant No. 24HASTIT048), Science and Technology Innovation Team Project of Zhengzhou University of Light Industry (Grant No. 23XNKJTD0101).

Author Contributions: Xiaoyu Wen: Methodology, conceptualization, writing—original draft, writing—review & editing, supervision. Haohao Liu: Methodology, data curation, validation, writing—original draft. Xinyu Zhang: Conceptualization, writing—review & editing. Haoqi Wang: Formal analysis, software, investigation. Yuyan Zhang: Methodology, investigation, validation. Guoyong Ye: Writing—review & editing, supervision. Hongwen Xing: Formal analysis, resources, investigation. Siren Liu: Data curation, resources, supervision. Hao Li: Project administration. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, [Hao Li], upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Zhang H. Introduction of aircraft assembly equipment and suppliers. Aeronaut Manuf Technol. 2008;51(11):71–3. (In Chinese). doi:10.3969/j.issn.1671-833X.2008.11.014.
2. Zhang XW, Lü RQ, Du XC, Zhang JE. Research on simulation and optimization of aircraft assembly pulsating production line. J New Ind. 2023;13(12):87–95. (In Chinese). doi:10.3969/j.issn.2095-6649.2023.12.009.
3. Blanco D, Rubio EM, Agustina B, Marín MM, Camacho AM. Advanced procedures and scheduling for aircraft assembly processes: a systematic review approach. Procedia CIRP. 2024;126:817–22. doi:10.1016/j.procir.2024.08.264.
4. Pan Z, Guo Y, Zha S, Zhang S, Wang B. Aircraft pulsating assembly line balancing problem based on hybrid algorithm. Comput Integr Manuf Syst. 2018;24(10):2436–47. Available from: http://www.cims-journal.cn/CN/10.13196/j.cims.2018.10.007.
5. Bao Z, Chen L, Qiu K. An aircraft final assembly line balancing problem considering resource constraints and parallel task scheduling. Comput Ind Eng. 2023;182(2):109436. doi:10.1016/j.cie.2023.109436.
6. Korivand S, Galvani G, Ajoudani A, Gong J, Jalili N. Optimizing human-robot teaming performance through Q-learning-based task load adjustment and physiological data analysis. Sensors. 2024;24(9):2817. doi:10.3390/s24092817.
7. Wen X, Zhang X, Xing H, Ye G, Li H, Zhang Y, et al. An improved genetic algorithm based on reinforcement learning for aircraft assembly scheduling problem. Comput Ind Eng. 2024;193(4):110263. doi:10.1016/j.cie.2024.110263.
8. Zheng Q. Research on assembly operation scheduling problem of aircraft final assembly moving production line with multiple parallel operations [dissertation]. Shanghai, China: Shanghai Jiao Tong University; 2015. (In Chinese).
9. Xin B, Li Y, Yu J, Zhang J. An adaptive BPSO algorithm for multi-skilled workers assignment problem in aircraft assembly lines. Assem Autom. 2015;35(4):317–28. doi:10.1108/aa-06-2015-051.
10. Ma'ruf A, Gultom SUAM. Multi-skilled operator assignment model for long cycle time and low-volume assembly line: a case study of aircraft assembly line. In: Proceedings of the 6th Asia Pacific Conference on Manufacturing Systems and 4th International Manufacturing Engineering Conference; 2022 Oct 27; Surakarta, Indonesia. Singapore: Springer Nature Singapore; 2023. p. 195–204. doi:10.1007/978-981-99-1245-2_18.
11. Wang P, Pei F, Liu J, Guo H, Zhuang C. Assembly scheduling problem of complex products considering multi-skill level worker assignment. J Mech Eng. 2025;4:389–402. (In Chinese).
12. Allemang–Trivalle A, Donjat J, Bechu G, Coppin G, Chollet M, Klaproth OW, et al. Modeling fatigue in manual and robot-assisted work for operator 5.0. IISE Trans Occup Ergon Hum Factors. 2024;12(1–2):135–47. doi:10.1080/24725838.2024.2321460.

13. Bruni ME, Beraldi P, Guerriero F, Pinto E. A heuristic approach for resource constrained project scheduling with uncertain activity durations. Comput Oper Res. 2011;38(9):1305–18. doi:10.1016/j.cor.2010.12.004.
14. Wang D, Qiao F, Guan L, Liu J, Ding C, Shi J. Human-machine collaborative optimization method for dynamic worker allocation in aircraft final assembly lines. Comput Ind Eng. 2024;194(3):110370. doi:10.1016/j.cie.2024.110370.
15. Chen S, Zhang N, Wang A. Pulsating assembly production personnel scheduling technology based on improved genetic algorithm. In: 2022 IEEE International Conference on Mechatronics and Automation (ICMA); 2022 Aug 7–10; Guilin, China. p. 1346–51. doi:10.1109/ICMA54519.2022.9856253.
16. Ding H, Zhuang C, Liu J. Extensions of the resource-constrained project scheduling problem. Autom Constr. 2023;153(14):104958. doi:10.1016/j.autcon.2023.104958.
17. Li X, He Z, Wang N. A branch-and-bound algorithm for the proactive resource-constrained project scheduling problem with a robustness maximization objective. Comput Oper Res. 2024;166(2):106623. doi:10.1016/j.cor.2024.106623.
18. Liu W, Zhang J, Liu C, Qu C. A bi-objective optimization for finance-based and resource-constrained robust project scheduling. Expert Syst Appl. 2023;231(1):120623. doi:10.1016/j.eswa.2023.120623.
19. Roussel S, Polacsek T, Chan A. Assembly line preliminary design optimization for an aircraft. In: The 29th International Conference on Principles and Practice of Constraint Programming; 2023 Aug 27–31; Toronto, ON, Canada. doi:10.4230/LIPIcs.CP.2023.32.
20. Long T, Li Y, Chen J. Productivity prediction in aircraft final assembly lines: comparisons and insights in different productivity ranges. J Manuf Syst. 2022;62:377–89. doi:10.1016/j.jmsy.2021.12.010.
21. Cai W, Zhao Y, Chen HJ, Zhang J. Improved genetic algorithm variable neighborhood search for solving aircraft assembly line scheduling problem. Manuf Autom. 2021;43(4):69–73,89. (In Chinese). doi:10.3969/j.issn.1009-0134.2021.04.015.
22. Shan S, Hu Z, Liu Z, Shi J, Wang L, Bi Z. An adaptive genetic algorithm for demand-driven and resource-constrained project scheduling in aircraft assembly. Inf Technol Manag. 2017;18(1):41–53. doi:10.1007/s10799-015-0223-7.
23. Ren Y, Lu Z. A flexible resource investment problem based on project splitting for aircraft moving assembly line. Assem Autom. 2019;39(4):532–47. doi:10.1108/aa-09-2018-0126.
24. Ren Y, Lu Z, Liu X. A branch-and-bound embedded genetic algorithm for resource-constrained project scheduling problem with resource transfer time of aircraft moving assembly line. Optim Lett. 2020;14(8):2161–95. doi:10.1007/s11590-020-01542-x.
25. Lu Z, Ren Y, Wang L, Zhu H. A resource investment problem based on project splitting with time windows for aircraft moving assembly line. Comput Ind Eng. 2019;135(21–22):568–81. doi:10.1016/j.cie.2019.06.044.
26. Borreguero Sanchidrián T, Portoleau T, Artigues C, García Sánchez A, Ortega Mier M, Lopez P. Large neighborhood search for an aeronautical assembly line time-constrained scheduling problem with multiple modes and a resource leveling objective. Ann Oper Res. 2024;338(1):13–40. doi:10.1007/s10479-023-05629-3.
27. Fang P, Yang J, Liao Q, Zhong RY, Jiang Y. Flexible worker allocation in aircraft final assembly line using multiobjective evolutionary algorithms. IEEE Trans Ind Inform. 2021;17(11):7468–78. doi:10.1109/TII.2021.3051896.
28. Arkhipov D, Battaïa O, Cegarra J, Lazarev A. Operator assignment problem in aircraft assembly lines: a new planning approach taking into account economic and ergonomic constraints. Procedia CIRP. 2018;76(2):63–6. doi:10.1016/j.procir.2018.01.020.
29. Ottogalli K, Rosquete D, Rojo J, Amundarain A, María Rodríguez J, Borro D. Virtual reality simulation of human-robot coexistence for an aircraft final assembly line: process evaluation and ergonomics assessment. Int J Comput Integr Manuf. 2021;34(9):975–95. doi:10.1080/0951192x.2021.1946855.
30. Tereshchuk V, Bykov N, Pedigo S, Devasia S, Banerjee AG. A scheduling method for multi-robot assembly of aircraft structures with soft task precedence constraints. Robot Comput Integr Manuf. 2021;71(15–16):102154. doi:10.1016/j.rcim.2021.102154.
31. Guo D. Fast scheduling of human-robot teams collaboration on synchronised production-logistics tasks in aircraft assembly. Robot Comput Integr Manuf. 2024;85(3):102620. doi:10.1016/j.rcim.2023.102620.

32. Duan Q, Liao TW. Improved ant colony optimization algorithms for determining project critical paths. Autom Constr. 2010;19(6):676–93. doi:10.1016/j.autcon.2010.02.012.
33. Gad AG. Particle swarm optimization algorithm and its applications: a systematic review. Arch Comput Meth Eng. 2022;29(5):2531–61. doi:10.1007/s11831-021-09694-4.
34. Wang F, Wang X, Sun S. A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. Inf Sci. 2022;602:298–312. doi:10.1016/j.ins.2022.04.053.
35. Guo WF, Song YC, Zhou F, Lei Q, Lyu XF. Integrated scheduling algorithm of complex product with no-wait constraint based on reversed virtual component. Comput Integr Manuf Syst. 2020;26(12):3313–28. (In Chinese). doi:10.13196/j.cims.2020.12.014.
36. Wang Q. Research on integrated scheduling algorithm driven by inverse sequence dynamics [dissertation]. Harbin, China: Harbin University of Science and Technology; 2023. (In Chinese). doi:10.27063/d.cnki.ghlgu.2023.001088.
37. Tanweer MR, Suresh S, Sundararajan N. Self regulating particle swarm optimization algorithm. Inf Sci. 2015;294(4):182–202. doi:10.1016/j.ins.2014.09.053.
38. Han H, Bai X, Han H, Hou Y, Qiao J. Self-adjusting multitask particle swarm optimization. IEEE Trans Evol Comput. 2022;26(1):145–58. doi:10.1109/TEVC.2021.3098523.
39. Arviv K, Stern H, Edan Y. Collaborative reinforcement learning for a two-robot job transfer flow-shop scheduling problem. Int J Prod Res. 2016;54(4):1196–209. doi:10.1080/00207543.2015.1057297.
40. Chalmers E, Contreras EB, Robertson B, Luczak A, Gruber A. Learning to predict consequences as a method of knowledge transfer in reinforcement learning. IEEE Trans Neural Netw Learn Syst. 2018;29(6):2259–70. doi:10.1109/TNNLS.2017.2690910.
41. Park KT, Jeon SW, Noh SD. Digital twin application with horizontal coordination for reinforcement-learning-based production control in a re-entrant job shop. Int J Prod Res. 2022;60(7):2151–67. doi:10.1080/00207543.2021.1884309.
42. Lu L, Zheng H, Jie J, Zhang M, Dai R. Reinforcement learning-based particle swarm optimization for sewage treatment control. Complex Intell Syst. 2021;7(5):2199–210. doi:10.1007/s40747-021-00395-w.
43. The operations research & scheduling research data; 2025. [cited 2025 Jan 1]. Available from: https://www.projectmanagement.ugent.be/research/data.