

ORIGINAL ARTICLE

International Journal of Applied Mathematics in Control Engineering

Journal homepage: <http://www.ijamce.com>

Research on Scheduling Optimization for Flexible Flow Shops with Tooling Matching Based on Hier-seq2seq

Kehua Wu^{1*} | Zhonghua Han¹

¹School of Electrical and control Engineering,
Shenyang Jianzhu University, Shenyang,
Liaoning, 110168, China

Correspondence

School of Electrical and control Engineering,
Shenyang Jianzhu University, Shenyang,
Liaoning, 110168, China
Email: wkh20010108@163.com

Article Info

Article history:

Received 12 March 2026

Accepted 18 May 2026

Available online 20 May 2026

Abstract

In the machining process, jobs require auxiliary processing with tooling. There is a many-to-many relationship between jobs and tooling, while tooling resources are limited and need to be reused across multiple production processes. If a job to be produced does not have available matching tooling, waiting time will occur, resulting in inefficient use of production resources. This paper models the scheduling optimization problem for flexible flow shops with tooling matching and proposes a scheduling optimization method based on a hierarchical sequence-to-sequence (Hier-seq2seq) model. The Seq2seq model is well suited for handling data with sequential characteristics and typically uses a single encoder. However, when dealing with data with complex matching relationships between multiple jobs and tooling, it is difficult to effectively capture the features and inherent relationships within the data. To address this, the paper constructs a hierarchical encoder. First, an operation-level encoder is used to extract the processing features of individual jobs, and then a workpiece-level encoder is applied to capture the global dependencies among jobs. This hierarchical encoder integrates local process information with global dependencies, ensuring the generation of high-quality job sequences. Taking the tooling matching rate as the optimization objective, simulation results demonstrate the proposed Hier-seq2seq-based scheduling optimization method can effectively improve the tooling matching rate, yielding good scheduling performance under different data scales.

KEYWORDS

Tooling matching; job sequence; scheduling optimization; seq2seq

1 | INTRODUCTION

Against the backdrop of modern manufacturing transitioning towards intelligence and customization, flexible flow shops have been widely used in semiconductor manufacturing[1], automotive parts processing, and heavy equipment production. Flexible flow shops require not only determining the processing sequence of jobs at each processing stage, but also assigning jobs to parallel machines[2]. Traditional flexible flow shop scheduling optimization focuses on constraints related to single-machine

resources. However, in actual production, the manufacturing of complex products relies not only on primary production resources such as machines and equipment, but also on auxiliary resources like tooling. In other words, processing can only be completed when both types of resources are simultaneously available[3]. Due to the current prevalence of small-batch and customized production, limited tooling resources can lead to poor matching between jobs and tooling, resulting in waiting times and wasted production resources [4]. As a result, operations involving tooling matching have become a key bottleneck restricting overall production efficiency in enterprises. Therefore, studying scheduling optimization for flexible flow shops with tooling matching has important practical and engineering value.

Rational scheduling optimization under limited tooling resources has attracted increasing attention. Considering tooling as auxiliary resources, Zhu et al. [5] constructed a parallel machine equally lot-sizing and scheduling model aimed at minimizing the maximum deviation between completion time and delivery time, and proposed a bi-objective whale swarm algorithm incorporating fast non-dominated sorting. To solve the parallel-machine scheduling problem considering mold constraints, Zhang et al.[6] developed a mathematical model to minimize the makespan and designed a hybrid differential genetic algorithm based on differential algorithms. To address the non-identical parallel-machine scheduling problem constrained by multiple types and numerous tooling resources, Zhou and Gu[7] established a mathematical model to minimize total weighted completion time and energy consumption, and proposed a multi-objective immune clone selection algorithm with elitist strategy. In scheduling problems considering limited fixture-pallet resources, Liu et al. [8] developed a mixed integer programming model with machine-fixture-pallet constraints to minimize the makespan, and proposed a novel genetic algorithm to optimize scheduling by solving the allocation of tooling across workstations. Kovalev et al. [9] discussed several results regarding the single-machine scheduling problems with auxiliary resource constraints, and proved that scheduling problems with tooling resource constraints are NP-hard. Lu et al. [10] considered mold constraints in the first and third stages and proposed a migratory algorithm to solve the problem. Juan C[11] proposed an enhanced metaheuristic algorithm to solve the parallel machine scheduling problem involving setup times and additional resources; Gao [12] presented a multi-objective evolutionary algorithm framework for similar problems. Wang et al.[13] investigated multi-objective scheduling methods under auxiliary resource constraints. Lei [14] proposed an adaptive artificial bee colony algorithm that utilizes a novel solution representation to solve the parallel machine scheduling problem. A review of recent literature shows scheduling problems in flexible flow shops with tooling matching are predominantly addressed using improved swarm intelligence and evolutionary algorithms. While these approaches can solve the problem to some extent, they typically involve complex iterative optimization processes and suffer from relatively low computational efficiency. Furthermore, existing studies have not thoroughly examined how complex matching relationships between jobs and tooling affect scheduling results.

In recent years, the application of deep learning algorithms has provided new research ideas for complex scheduling problems. Han et al. [15] have applied deep learning algorithms to solve flexible flow shop scheduling problems. This approach has shown advantages in both computational efficiency and generalization capability. This paper employs deep learning methods to the scheduling optimization problem of flexible flow shops with tooling matching, proposing a scheduling optimization method based on Hier-seq2seq model. By using a hierarchical encoder structure, the model enhances its ability to capture relationships between jobs and tooling, as well as their impact on scheduling outcomes, thereby obtaining a better job sequence. This approach provides an efficient and generalizable solution for flexible flow shop scheduling problems with tooling matching.

2 | MATHEMATICAL MODEL

2.1 | Problem Description

The flexible flow shop scheduling optimization problem with tooling matching is described as follows. In actual production, once a production order is issued, there are N_j jobs to be processed. They are released onto the production line in a predetermined order and undergo processing through N_{pro} stages. Each stage Pro_j consists of a set of parallel machines with identical functions but varying quantities, where each machine represents a workstation. Each workstation can only process one job at a time. The production process of flexible flow shops with tooling matching is shown in Figure 1. After completing processing at stages

Pro_{bf} that do not require tooling assistance, jobs enter the buffer of stages that require tooling assistance. If a workstation at a tooling-assisted stage becomes idle, but no job in the buffer matches the available tooling, or no tooling matches the job, the process must wait until a compatible job or tooling becomes available. If both a job and its matching tooling are available in the buffer, the job can proceed to be processed at tooling-assisted stage. The job then continues through Nfp tooling-assisted stages. At stage $Pro_{bf+Nfp-1}$, the tooling is removed and returned to the buffer of stage Pro_{bf} , where tooling assistance is required, enabling reuse of tooling. After tooling removal, the job continues the remaining production processes. Taking tooling matching rate as the optimization objective, an intelligent optimization algorithm is employed to generate scheduling results. These results include the assigned workstation, start time, and completion time of each job at each stage, as well as the usage status of tooling during processing.

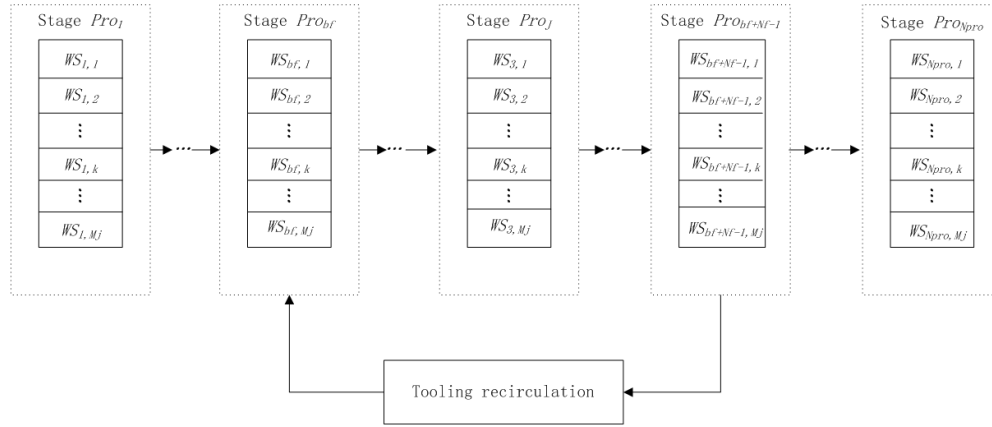


FIGURE 1 Production process diagram of a flexible flow shop with tooling matching.

2.2 | Model Parameters

To describe the elements and their relationships in the flexible flow shop scheduling optimization model with tooling matching, the following parameter symbols are defined:

Nj : total number of jobs.

$Npro$: Total number of processing stages.

$Proj$: The j -th stage, where $j \in \{1, 2, \dots, Npro\}$.

$Lpro$: Set of stages, $Lpro = \{Pro_1, Pro_2, \dots, Proj, \dots, Pro_{Npro}\}$.

Pro_{bf} : The first stage assisted by tooling.

Nfp : Total number of tooling-assisted stages.

$Lprofp$: Set of tooling-assisted stages, $Lprofp = \{Pro_{bf}, Pro_{bf+1}, \dots, Pro_{bf+Nfp-1}\}$, $Lprofp \in Lpro$.

Kj_k : The k -th job type, where $k \in \{1, 2, \dots, Nkj\}$, and Nkj denotes the total number of job types.

Lkj : Set of job types, $Lkj = \{Kj_1, Kj_2, \dots, Kj_{Nkj}\}$.

$J_{i,k}$: The i -th job of job type Kj_k , where $i \in \{1, 2, \dots, Nj\}$, $k \in \{1, 2, 3 \dots Nkj\}$, is regarded as the minimum processing task unit.

$WS_{j,m}$: The m -th workstation of stage $Proj$, where $m \in \{1, 2, \dots, Mj\}$, and Mj denotes the maximum number of workstations included at stage $Proj$.

Ft_g : The g -th tooling, where $g \in \{1, 2, \dots, Ng\}$.

Kf_z : The z -th tooling type, where $z \in \{1, 2, 3 \dots Nkf\}$, and Nkf denotes the total number of tooling types.

Lkf : Set of tooling types, $Lkf = \{Kf_1, Kf_2, \dots, Kf_{Nkf}\}$.

$Rkmf_{g,z}$: Correspondence relationship between tooling type and tooling, $Rkmf = (Ft_g, Kf_z)$.

$Rkfmkj_{z,k}$: Matching relationship between tooling type and job type, $Rkfmkj_{z,k} = (Kf_z, Kj_k)$.

$TS_{i,k,j}$: Start time of job $J_{i,k}$ of type Kj_k at stage Pro_j .

$TC_{i,k,j}$: Completion time of job $J_{i,k}$ of type Kj_k at stage Pro_j .

$TSp_{k,j}$: Processing time (time period, duration) of job of type Kj_k at stage Pro_j .

2.3 | Constraints

Based on the model parameters defined above, mathematical equations are established to describe the constraint relationships among variables during the production, thereby reflecting the process and characteristics of the flexible flow shop scheduling problem with tooling matching.

(1) General Constraints for Flexible Flow Shop Scheduling Optimization Problem with Tooling Matching

$$TC_{i,k,j} = TS_{i,k,j} + TSp_{k,j}, i \in \{1, 2, \dots, Njob\}, j \in \{1, 2, \dots, Npro\}, k \in \{1, 2, 3 \dots Nkj\} \quad (1)$$

Equation (1) indicates that the completion time $TC_{i,k,j}$ of job $J_{i,k}$ of type Kj_k at stage Pro_j is equal to start time $TS_{i,k,j}$ plus the processing time $TSp_{k,j}$.

$$TS_{i,k,j+1} \geq TC_{i,k,j}, j \in \{1, \dots, Npro - 1\} \quad (2)$$

Equation (2) means the next stage Pro_{j+1} of job $J_{i,k}$ of type Kj_k can only start after completing the current stage Pro_j .

(2) Tooling Occupancy and Availability Constraints

Equation (5) indicates the quantity of available tooling of type Kf_z at time t , denoted as $Qvf_z(t)$, is equal to the total number of tooling of type Kf_z , denoted as Qf_z , minus the number of tooling of the same type currently being used at all stages assisted by tooling at time t . The stages in Equations (4) and (5) satisfy $Pro_j \in Lprofp$.

$$Qf_z = \text{card}(\{Rkmf_{g,z}\}) \quad (3)$$

$$AQfu_{i,k,z,j}(t) = \begin{cases} 1 & \text{At time } t, \text{ job } J_{i,k} \text{ uses a tooling of type } Kf_z \text{ at stage } Pro_j \\ 0 & \text{At time } t, \text{ job } J_{i,k} \text{ does not use a tooling of type } Kf_z \text{ at stage } Pro_j. \end{cases} \quad (4)$$

$$Qvf_z(t) = Qf_z - \sum_{i=1}^{Njob} \sum_{j=1}^{Npro} AQfu_{i,k,z,j}(t) \quad (5)$$

In Equation (6), $Qvfkfjk_k(t)$ represents the number of tooling of type Kf_z that match the job type Kj_k of job $J_{i,k}$ at time t . The time t falls within the interval between the completion time $TC_{i,k,bf-1}$ of job $J_{i,k}$ at the previous stage Pro_{bf-1} , and the start time $TS_{i,k,bf}$ of job $J_{i,k}$ at the first tooling-assisted stage Pro_{bf} , namely $t \in [TC_{i,k,bf-1}, TS_{i,k,bf}]$.

At time t , for job $J_{i,k}$ of type Kj_k at stage Pro_j , if $Qvfkfjk_k(t) > 0$, tooling-assisted production can be conducted. If $Qvfkfjk_k(t) = 0$, job $J_{i,k}$ cannot be processed and must wait for the matching tooling to return. The stage satisfies $Pro_j \in Lprofp$.

$$Qvfkfjk_k(t) = \prod_{z=1}^{Nkf} (\{Qvf_z(t) | Rkfmjk_{z,k}\}) \quad (6)$$

2.4 | Scheduling Optimization Objective

When job $J_{i,k}$ enters the first stage Pro_{bf} using tooling-assisted production, its matching tooling Ft_g is still in use. In this case, job $J_{i,k}$ must wait until its matching tooling is released at the last stage $Pro_{bf+Nfp-1}$ using tooling-assisted production and then returned to the tooling-installed stage Pro_{bf} . If this occurs, it is recorded as a tooling matching delay for this job, meaning that the tooling is not matched in time. By counting the number of jobs experiencing matching delays, the tooling matching rate $Rmatch$ is designed, as shown in Equation 8.

Here, $JobWaitFix_i$ is used to indicate whether job $J_{i,k}$ experiences a tooling matching delay. Specifically, within the interval between the completion time $TC_{i,k,bf-1}$ of job $J_{i,k}$ at the previous stage Pro_{bf-1} , and the start time $TS_{i,k,bf}$ of job $J_{i,k}$ at the first

tooling-assisted stage Pro_{bf} , if the number of tooling of type Kf_z matching the job type Kj_k , denoted as $Qvkfmkj_k(t)$, equals 0, then a matching delay is recorded and $JobWaitFix_i = 1$. If $Qvkfmkj_k(t) \neq 0$, then no matching delay occurs and $JobWaitFix_i = 0$.

$$JobWaitFix_i = \begin{cases} 1 & \exists Qvkfmkj_k(t) = 0, t \in [TC_{i,k,bf-1}, TS_{i,k,bf}] \\ 0 & \nexists Qvkfmkj_k(t) = 0, t \in [TC_{i,k,bf-1}, TS_{i,k,bf}] \end{cases} \quad (7)$$

$$Rmatch = \left(1 - \frac{\sum_{i=1}^{Njob} JobWaitFix_i}{Njob} \right) \times 100\% \quad (8)$$

3 | DEEP LEARNING MODEL DESIGN FOR SCHEDULING OPTIMIZATION PROBLEMS WITH TOOLING MATCHING

3.1 | Application and Improvement of Sequence-to-Sequence Models

The optimization of job sequence is essentially a complex sequence generation problem, and its resulting scheduling scheme exhibits strong sequential characteristics. Seq2seq models excel at handling such sequential data. After training, they can generate optimal job sequences, enabling end-to-end fast inference. Traditional single encoders, however, typically only capture temporal information of jobs and struggle to learn how job-tooling matching relationships influence scheduling results. Although they remain applicable to general scheduling problems, they are insufficient for handling complex scenarios involving tooling matching. To address the flexible flow shop scheduling problem with tooling matching, this paper proposes a Hier-seq2seq-based scheduling optimization method. Its core idea is to construct a hierarchical encoder within the hierarchical sequence-to-sequence framework. The operation encoder extracts the processing attributes of each job at different stages and their specific tooling requirements. The workpiece encoder captures global dependencies arising from tooling competition due to job-tooling matching relationships. The workpiece vectors are then weighted and aggregated into a context vector c , which is used to initialize the decoder's hidden state, thereby injecting global information into the sequence generation process. During training, the decoder uses the ground-truth value from the previous time step of the target sequence as its input. During inference, the model takes a start token as the initial input and, at each time step, uses its previous prediction as the current input. Finally, a greedy strategy is applied at each time step to select the job with the highest probability as the current output, until a complete optimized job sequence is generated. The structure of the proposed Hier-seq2seq model for solving the flexible flow shop scheduling problems with tooling matching is shown in Figure 2.

3.2 | A Scheduling Optimization Method Based on Hier-seq2seq Model

This paper proposes a scheduling optimization method based on Hier-seq2seq model for flexible flow shops with tooling matching. The method primarily utilizes the Hier-seq2seq model to optimize the job sequence. The overall process of this method is shown in Figure 3. The simulation data are derived from historical production and processing data collected from a manufacturing flexible flow shop. The source sequences in the dataset include attributes such as the processing time at each stage for each job, the types of matched tooling, and basic job information. The target sequences correspond to the optimized job sequences. First, the historical data are preprocessed through numerical representation, normalization, and data encoding. The processed dataset is then divided into training, validation, and test datasets in a ratio of 7:2:1 and fed into the model. During the training phase, the model-generated job sequences are compared with historically optimized production data to adjust model parameters. After training is complete, the model can output valid scheduling results from a single scheduling task input.

4 | SIMULATION EXPERIMENTS

The algorithm was implemented in Python 3.0, using the PyTorch deep learning framework. The experiments were conducted on a Windows 11 operating system. The hardware platform was configured with an AMD Ryzen 9950X CPU, an AMD Radeon

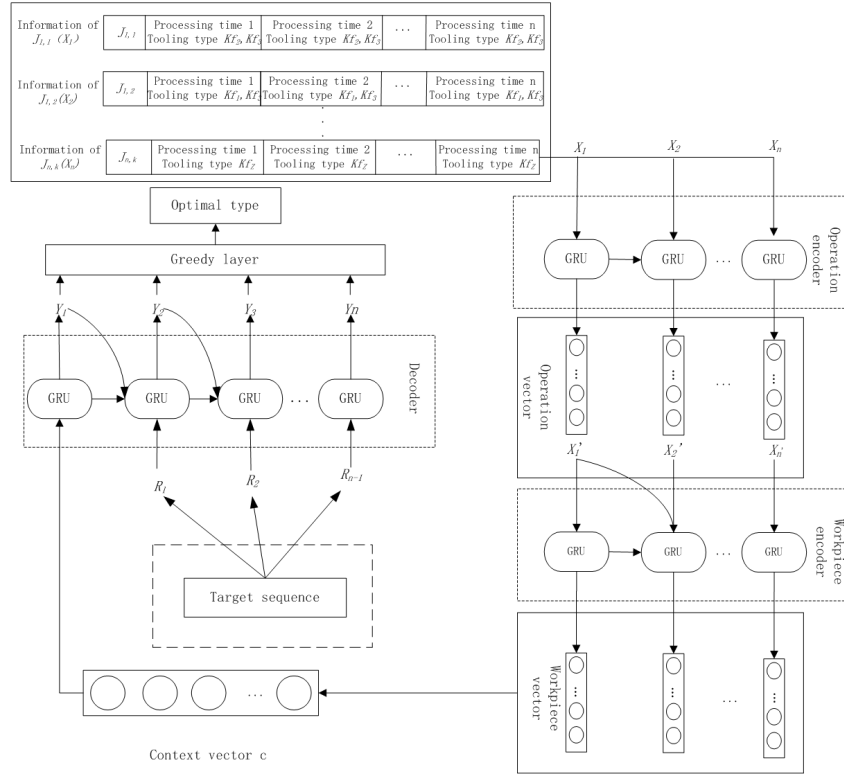


FIGURE 2 Hier-seq2seq structure diagram

RX 9070 XT GPU, and 64GB of memory. Experimental data came from a flexible flow shop in a large manufacturing enterprise, with scale and parameters derived from actual production. To better analyse and evaluate scheduling results, makespan fc and minimum completion time were used as evaluation metrics, combined with tooling matching rates $Rmatch$.

In the flexible flow shop scheduling optimization problem, makespan fc is a classical indicator for measuring the overall efficiency of a production system. It is defined as the maximum completion time $TC_{i,k,Npro}$ of all jobs at the last stage Pro_{Npro} , as shown in Equation 9.

$$fc = \max \{ TC_{i,k,Npro} \} \quad (9)$$

4.1 | Analysis of Model Structure Effectiveness

This paper compares the standard Seq2seq model with the Hier-seq2seq model. Both models use the same recurrent neural network in the encoder and decoder components, while the Hier-seq2seq encoder is enhanced with a hierarchical encoder structure. Their loss functions are compared in Figure 4.

The experimental results are shown in the figure. As can be observed from Figure 4, after 300 training epochs, the loss values of both methods begin to converge. The Hier-seq2Seq model achieves a faster loss reduction rate in the early stages of training and reaches a lower stable loss value in the later stages, while the Seq2seq model's stable loss value is markedly higher than that of Hier-seq2seq. This indicates that Hier-seq2seq has better learning capabilities.

5 | SIMULATION VERIFICATION ANALYSIS

To verify the effectiveness of the proposed method in a real production environment, the flow shop production information was established as shown in Table 1. The scheduling task includes a total of 36 jobs to be processed ($Nj=36$), consisting of 8 job types. The processing time and required tooling types for each job are presented in Table 1. The production line has limited

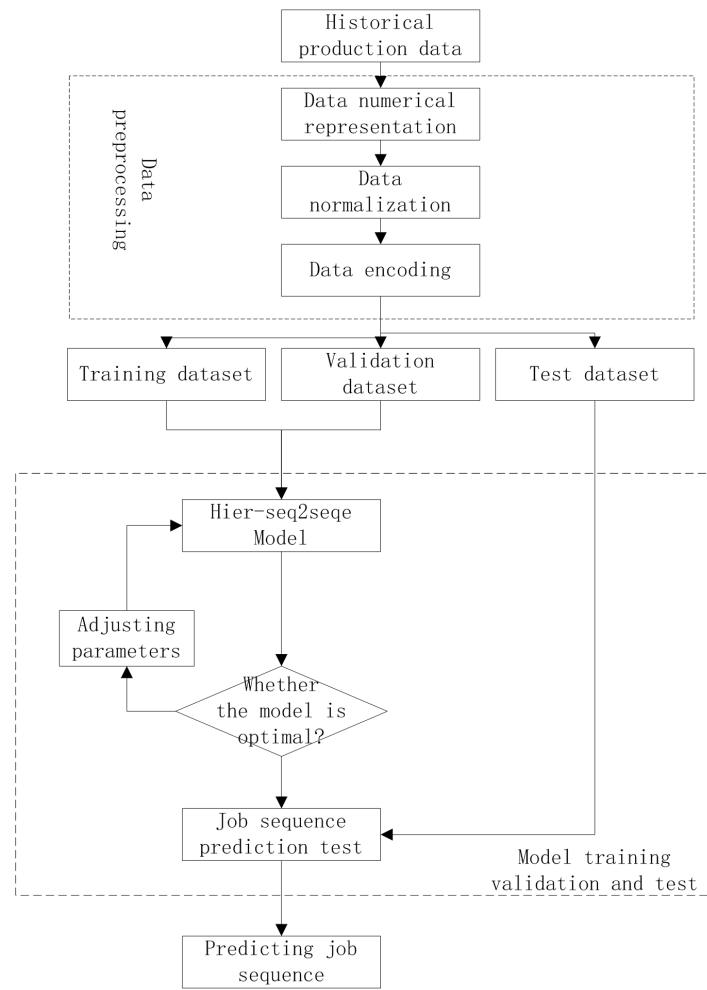


FIGURE 3 Flowchart of the Hier-seq2seq-based scheduling optimization method

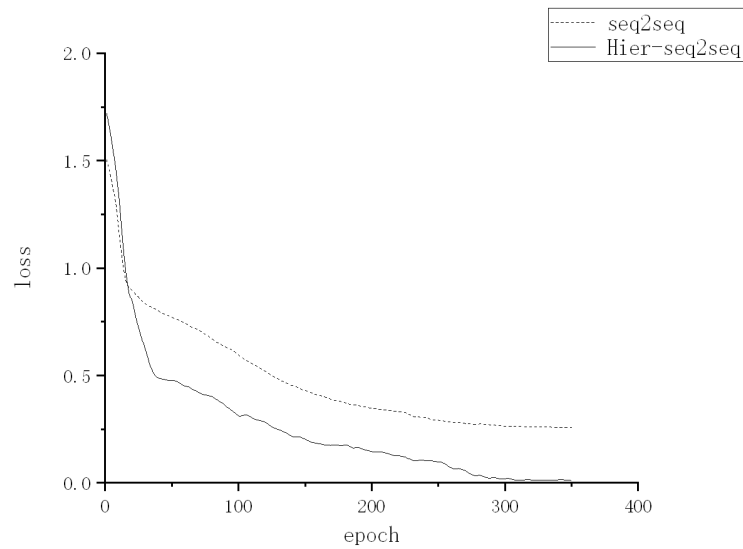


FIGURE 4 Loss versus training epochs plot

tooling resources and exhibits an obvious matching characteristic.

It includes 3 sets of tooling type $Kf_1 : \{Rkmf_{1,1}, Rkmf_{2,1}, Rkmf_{3,1}\} = \{(Ft_1, Kf_1), (Ft_2, Kf_1), (Ft_3, Kf_1)\}$, 4 sets of tooling type $Kf_2 : \{Rkmf_{4,2}, Rkmf_{5,2}, Rkmf_{6,2}, Rkmf_{7,2}\} = \{(Ft_4, Kf_2), (Ft_5, Kf_2), (Ft_6, Kf_2), (Ft_7, Kf_2)\}$, 3 sets of tooling type $Kf_3 : \{Rkmf_{8,3}, Rkmf_{9,3}, Rkmf_{10,3}\} = \{(Ft_8, Kf_3), (Ft_9, Kf_3), (Ft_{10}, Kf_3)\}$.

TABLE 1 Job Process Parameters and Tooling Matching Relationships

Job Type K_{j_k}	Processing Time $\{TSp_{k,j}\}$ (Unit: hour)	Matching Relationship $\{Rkfmk_{j_z,k}\}$	Quantity
K_{j_1}	$\{1.5, 1, 1.75, 1.26, 2, 1.3, 1.7\}$	$\{(Kf_2, K_{j_1}), (Kf_3, K_{j_1})\}$	3
K_{j_2}	$\{1.75, 1.4, 1.63, 1.87, 2.1, 1.23, 1.6\}$	$\{(Kf_1, K_{j_2}), (Kf_3, K_{j_2})\}$	1
K_{j_3}	$\{2, 1.47, 1.65, 1.77, 1.65, 1.23, 2\}$	$\{(Kf_2, K_{j_3})\}$	5
K_{j_4}	$\{1.23, 1.22, 2.1, 2.3, 1.35, 1.46, 1.45\}$	$\{(Kf_1, K_{j_4}), (Kf_2, K_{j_4})\}$	3
K_{j_5}	$\{1.45, 1.65, 1.78, 1.97, 1.45, 1.65, 1.24\}$	$\{(Kf_2, K_{j_5}), (Kf_3, K_{j_5})\}$	6
K_{j_6}	$\{2, 1.45, 2.1, 2.23, 2.45, 2.1, 2\}$	$\{(Kf_3, K_{j_6})\}$	10
K_{j_7}	$\{1.2, 2.1, 1.24, 2.3, 2.1, 1.4, 1.6\}$	$\{(Kf_1, K_{j_7})\}$	3
K_{j_8}	$\{1.4, 1.64, 1.24, 1.65, 1.34, 1.75, 1.65\}$	$\{(Kf_1, K_{j_8}), (Kf_2, K_{j_8})\}$	5

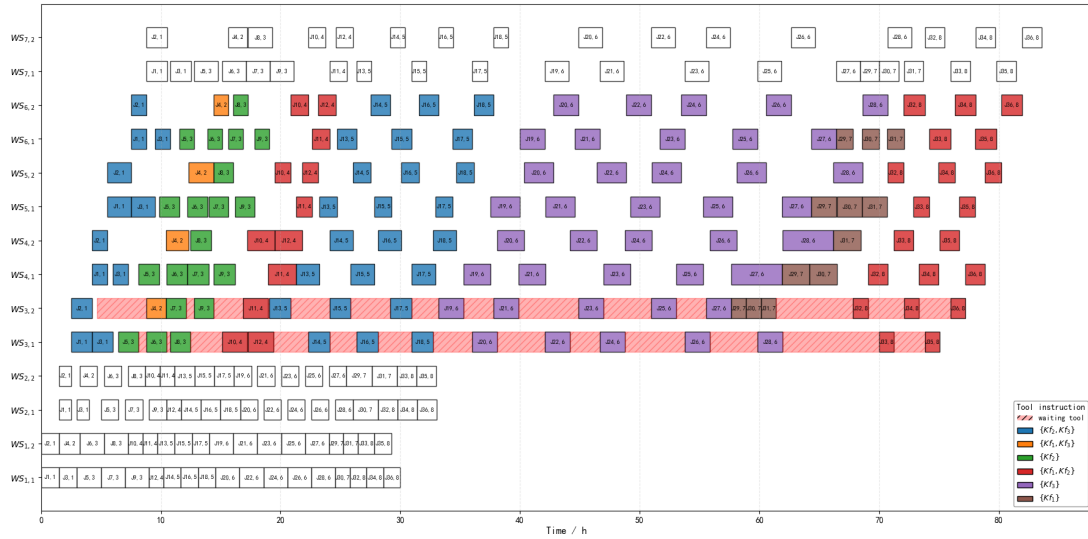


FIGURE 5 Gantt chart of scheduling results based on active scheduling algorithm

In Figures 5 and 6, the horizontal axis of the Gantt chart represents time, while the vertical axis indicates the workstation at each stage. Red diagonal lines represent the waiting time of jobs at the tooling installation station, i.e., delays caused by tooling mismatch. Jobs with the same tooling are assigned the same color.

Figure 5 is a Gantt chart of scheduling results obtained using the active scheduling algorithm. After completing its preceding operations, job $J_{4,2}$ enters workstation $WS_{3,2}$ of stage Pro_3 . At this time, tooling Kf_1 and tooling Kf_3 are needed for auxiliary processing. However, tooling Kf_3 is occupied by three earlier jobs ($J_{1,1}, J_{2,1}, J_{3,1}$) that require its assistance for auxiliary processing and remains unavailable. Therefore, job $J_{4,2}$ is stuck at stage Pro_3 , preventing subsequent jobs from entering the production line. The waiting time for jobs such as $J_{26,6}$, $J_{32,8}$ and $J_{33,8}$ all exceed 5 hours, forming a severe resource bottleneck. This issue occurs frequently throughout the entire production. Under this condition, the makespan $\overline{fC}$ for this order is 86.92 hours, and the tooling matching rate $Rmatch$ is only 21.1%, indicating significant production delays caused by tooling mismatch.

Figure 6 shows the Gantt chart of scheduling results obtained using a scheduling optimization method based on Hier-seq2seq. Due to a reasonable job sequence, frequent short-term overuse of identical tooling has been reduced. The red waiting-time areas become more fragmented and shorter, and most jobs' waiting time are reduced to under 5 hours. The job processing distribution is

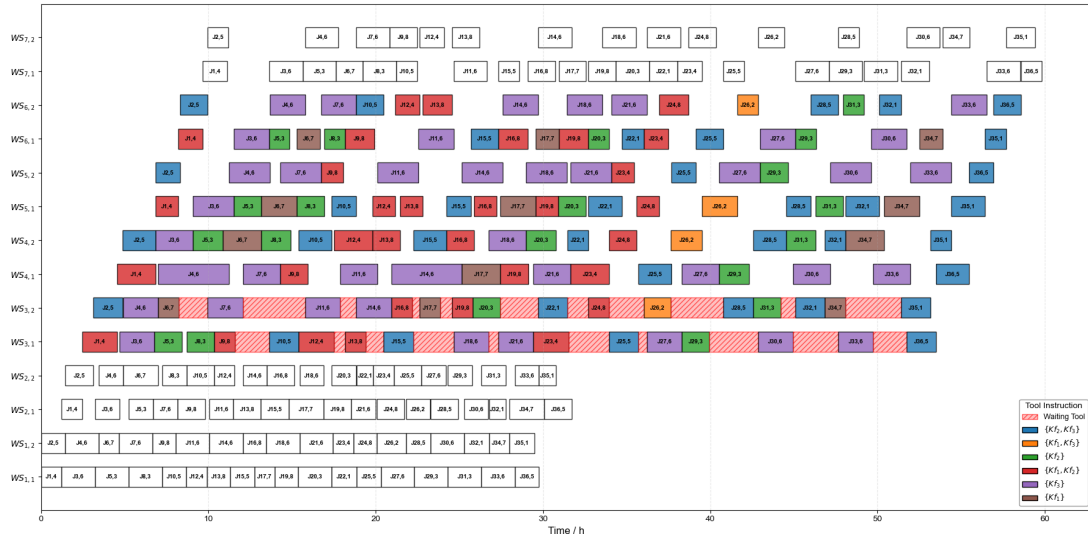


FIGURE 6 Gantt chart of scheduling results based on Hier-Seq2Seq

more compact, and idle time at workstations is significantly shortened. As a result, the makespan for this order is reduced to 58.2 hours, and the tooling matching rate R_{match} increases to 62.4%.

Therefore, the Hier-seq2seq-based scheduling optimization method can reduce the makespan fC , significantly improve the tooling matching rate R_{match} , reduce resource waste, and enhance overall production efficiency.

5.1 | Simulation Scheme Design

To systematically verify the effectiveness, stability, and generalization ability of the proposed Hier-seq2seq-based scheduling optimization method for solving the flexible flow shop scheduling problem with tooling matching, multiple groups of simulation experiments were designed for comprehensive analysis from different problem scales and comparative algorithms.

Four simulation schemes were designed: Scheme 1 uses a Genetic Algorithm (GA), Scheme 2 uses a Simulated Annealing (SA) algorithm, Scheme 3 uses a Variable Neighbourhood Search (VNS) algorithm, and Scheme 4 uses a Hier-seq2seq model. The number of algorithm iteration is set to 500. The details of simulation schemes are shown in Table 1.

TABLE 2 Simulation Scheme Table

Scheme	Scheduling Algorithm
Scheme 1	GA
Scheme 2	SA
Scheme 3	VNS
Scheme 4	Hier-seq2Seq

For the four simulation schemes, multiple simulation experiments were conducted with different production scales, thereby generating multiple sets of simulation cases. The average tooling matching rate $\overline{R_{match}}$ is used as the optimization objective, and the average makespan $\overline{fC}$ is used as the evaluation metric. This approach fully verifies the effectiveness and generalization performance of the proposed method. The simulation results are shown in Table 2.

As shown in Table 2, when the total number of jobs is small ($N_j = 10$), tooling resources are relatively sufficient on the production line. Therefore, there is little congestion caused by tooling shortages during scheduling, and the differences in

TABLE 3 Simulation Results for Different Scales

Number of jobs (Nj)	Indicator Mean	Scheme 1	Scheme 2	Scheme 3	Scheme 4
$Nj = 10$	$\overline{fC}$	27.66	26.33	26.42	25.33
$Nj = 10$	$\overline{Rmatch}$	91.22%	92.22%	91.12%	94.87%
$Nj = 30$	$\overline{fC}$	69.22	67.24	67.38	50.11
$Nj = 30$	$\overline{Rmatch}$	51.66%	50.32%	50.88%	62.53%
$Nj = 50$	$\overline{fC}$	93.33	92.11	92.43	81.22
$Nj = 50$	$\overline{Rmatch}$	41.43%	43.77%	45.32%	60.22%
$Nj = 100$	$\overline{fC}$	167.33	162.55	163.43	145.22
$Nj = 100$	$\overline{Rmatch}$	30.43%	31.77%	40.32%	55.22%

optimization performance among different scheduling methods are not obvious. At this stage, Schemes 1, 2, and 3 can maintain a relatively high tooling matching rate $\overline{Rmatch}$. However, Scheme 4 still shows a certain advantage in scheduling performance. Its average makespan $\overline{fC}$ is 25.33, which is 8.4% lower than Scheme 1, and its average tooling matching rate $\overline{Rmatch}$ reaches 94.87%, representing a 3.6% improvement over Scheme 1. Experimental results show that even with fewer tooling matching conflicts, the proposed method can still improve the overall production efficiency of the flexible flow shop by optimizing the job sequence.

As the number of jobs increases to $Nj = 30$, tooling shortages caused by tooling occupancy becomes more pronounced on the production line. Situations where jobs wait for available tooling occur frequently, significantly increasing the complexity of scheduling problems. Under this scenario, the performance of traditional scheduling methods degrades noticeably, with Schemes 1-3 showing substantial performance drops and tooling matching rate $\overline{Rmatch}$ falling to about 50%. Waiting times caused by tooling mismatches increase significantly on the production line, thus extending the overall production cycle. In contrast, Scheme 4 maintains a tooling matching rate $\overline{Rmatch}$ of 62.53%, significantly higher than the other schemes. Its average makespan $\overline{fC}$ is only 50.11, about 27.6% lower than that of Scheme 1. This demonstrates the Hier-seq2seq-based method can effectively alleviate production congestion caused by tooling conflicts in medium-scale scenarios.

As the number of jobs further increases to $Nj = 50$, tooling resources become a critical bottleneck for scheduling optimization. The tooling matching rates $\overline{Rmatch}$ of Schemes 1, 2, and 3 drop to 30.43%, 31.77%, and 40.32%, respectively. Although some methods alleviated resource conflicts to some extent, the overall tooling matching rate $\overline{Rmatch}$ still showed a downward trend, indicating their limited adaptability to complex resource constraints. In contrast, Scheme 4 maintains an average tooling matching $\overline{fC}$ rate of 60.22% and an average makespan $\overline{fC}$ of 81.22, performing best among all schemes. The results show that under medium-to-large-scale conditions, the Hier-seq2seq model still possesses good scheduling stability.

In a large-scale testing scenario ($Nj = 100$), the above trend is further confirmed. As production scale expands, the average of makespan $\overline{fC}$ of each scheme increases significantly. However, Scheme 4 still clearly outperformed the others, with an average makespan $\overline{fC}$ of 145.22 and a tooling matching rate $\overline{Rmatch}$ of 55.22%, demonstrating strong generalization ability.

The experimental results across different scales show that as the number of jobs increases, when $Nj \geq 30$, the tooling matching rates $\overline{Rmatch}$ of Schemes 1-3 show a clear downward trend, while Scheme 4 maintains a relatively high level. This indicates that in large-scale production environments, tooling conflicts have gradually evolved from local random phenomena to systematic resource congestion problems caused by job sequence. Traditional rule-based algorithms or conventional Seq2Seq models are difficult to explicitly capture the continuous occupancy relationships of tooling across multiple stages. However, the proposed Hier-seq2seq method, through hierarchical modelling of jobs and stages, can more effectively coordinate tooling resources as production scales expand, thus achieving better scheduling performance and strong generalization ability at different scales.

CONCLUSION

To address scheduling problems in flexible flow shops with tooling matching, this paper proposes a scheduling optimization method based on Hier-seq2seq. The model learns from historically optimized data to efficiently predict job sequences. Simulation results show that, compared with active scheduling algorithms, the proposed method reduces the makespan $\overline{fC}$ from 86.92 hours to 58.2 hours, and improves the tooling matching rate $\overline{Rmatch}$ from 21.1% to 62.4%. In medium-to-large-scale testing scenarios, this method outperforms swarm intelligence optimization algorithms in both scheduling efficiency and solution feasibility, demonstrating strong generalization ability.

This study verifies the effectiveness of deep learning in solving flexible flow shops scheduling problems with tooling matching. Future research could explore the application of reinforcement learning to solve such scheduling problems, and investigate the adaptability of the proposed method under dynamic disturbances such as emergency order insertion.

REFERENCES

- [1] Yu B. Review on flexible flow shop scheduling. *Modern Manufacturing Engineering*. FlatChem 2025;51:100862.
- [2] Ci X, Zhao W, Luo J, Wu Y, Ge T, Xue Q, et al. How the fluorographene replaced graphene as nanoadditive for improving tribological performances of GTL-8 based lubricant oil. *Friction* 2021;9(3):488–501.
- [3] Yongzhen S, Hanlu Z, Yelin S, Jingjuan J. Dynamic Coupling Analysis of High-Speed Railway and Remote Investment Network in China. *Tropical Geography* 2024;44(7).
- [4] Hao B, Y W, Xu X, Guo J. Review of AUV Underwater Recovery Docking Studies. *International Journal of Applied Mathematics in Control Engineering* 2024;7(1):41–47.
- [5] Cui J, Ma H, Gong S, Wang D, Zhang S, Liu Z, et al. Water-dispersible fluorinated graphene/MXene with stable heterostructure for responsive lubrication and long-term anticorrosion. *Tribology International* 2025;p. 110850.
- [6] Xue Z, Lv D, Du Y, Huo D. Digital Twin of Permanent Magnet Synchronous Motor Based on Finite Element Analysis and UNITY. *International Journal of Applied Mathematics in Control Engineering* 2024;7(2):156–164.
- [7] Nair RR, Ren W, Jalil R, Riaz I, Kravets VG, Britnell L, et al. Fluorographene: a two-dimensional counterpart of Teflon. *small* 2010;6(24):2877–2884.
- [8] Zbořil R, Karlický F, Bourlinos AB, Steriotis TA, Stubos AK, Georgakilas V, et al. Graphene fluoride: a stable stoichiometric graphene derivative and its chemical conversion to graphene. *small* 2010;6(24):2885–2891.
- [9] Yan X, Zhang X, Lv B, Zou Y, Zhang M, We H, et al. Atmospheric Switching between Etching and High-Quality Fluorination of Graphene by XeF₂. *ACS Applied Materials & Interfaces* 2025;17(21):31257–31264.
- [10] Liang L, Song L, Yang Y, Li F, Ma Y. Tribological properties of polytetrafluoroethylene improved by incorporation of fluorinated graphene with various fluorine/carbon ratios under dry sliding condition. *Tribology Letters* 2021;69(1):21.
- [11] Jahanshahi M, Kowsari E, Haddadi-Asl V, Khoobi M, Bazri B, Aryafard M, et al. An innovative and eco-friendly modality for synthesis of highly fluorinated graphene by an acidic ionic liquid: Making of an efficacious vehicle for anti-cancer drug delivery. *Applied Surface Science* 2020;515:146071.
- [12] Xu JY, Yu JS, Liao JH, Yang XB, Wu CY, Wang Y, et al. Opening the band gap of graphene via fluorination for high-performance dual-mode photodetector application. *ACS applied materials & interfaces* 2019;11(24):21702–21710.
- [13] Wang X, Dai Y, Gao J, Huang J, Li B, Fan C, et al. High-yield production of highly fluorinated graphene by direct heating fluorination of graphene-oxide. *ACS applied materials & interfaces* 2013;5(17):8294–8299.
- [14] Wang D, Jia X, Yang J, Wang Z, Song H. Fluorinated graphene intercalated by carbon quantum dots: Triple effect for reducing friction and wear. *Chemical Engineering Journal* 2025;512:162393.
- [15] Tian R, Jia X, Yang J, Li Y, Song H. Large-scale, green, and high-efficiency exfoliation and noncovalent functionalization of fluorinated graphene by ionic liquid crystal. *Chemical Engineering Journal* 2020;395:125104.



Kehua Wu is currently pursuing a master's degree at the School of Electrical and Control Engineering, Shenyang Jianzhu University. His primary research interests include production scheduling optimization and deep learning.



Zhonghua Han earned his Master's in Control Theory and Control Engineering from Shenyang Jianzhu University (SJZU), Shenyang, China in 2004, and PhD in Control Theory and Control Engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China in 2014. He is currently a Professor at the School of Electrical and Control Engineering, SJZU. He mainly engages in theoretical research and engineering applications of production optimisation, develops multiple large-scale manufacturing enterprise manufacturing control platforms.