U.P.B. Sci. Bull., Series D, Vol. 87, Iss. 3, 2025 ISSN 1454-2358

SCHEDULING OPTIMIZATION OF AEROSPACE REDUCER MACHINING OPERATIONS BASED ON DYNAMIC PROGRAMMING ALGORITHM

Chenghao PAN [1], Yifei TONG [2*], Tiefeng ZHANG [3]

As a key component of the transmission system, aerospace reducers are widely used in aerospace engines, rudders and other components. To address the extended completion times in small-batch, multi-process production of aerospace reducer shaft components, this paper proposes a scheduling optimization method based on a state-transfer dynamic programming algorithm. First, a mathematical model is established with the objective of minimizing the maximum completion time (makespan). Secondly, the model is solved using the proposed dynamic programming algorithm. The method is validated using an actual case from an aviation enterprise. Experimental results demonstrate a 13% reduction in makespan compared to the enterprise's original scheduling plan. Furthermore, a comparative analysis with a Genetic Algorithm reveals that the proposed method holds significant advantages in both computational efficiency and solution optimality, thereby validating its effectiveness.

Keywords: Aerospace reducer, Shafting components, Dynamic Programming, Scheduling optimization

1. Introduction

In recent years, with the rise of new technological revolutions driven by major technologies such as the Internet of Things [1], cloud manufacturing [2], and big data [3], the pace of global manufacturing transformation has accelerated.

[1] School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

[2*] Prof., School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, e-mail: tyf1776@mail.njust.edu.cn (Corresponding author)

[3] School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Represented by Germany's "Industry 4.0" plan, manufacturing powers and regions such as the United States (Industrial Internet), Japan (Industrial Value Chain Plan IVI) [4], and the European Union have vigorously promoted the manufacturing revolution with intelligent manufacturing as the core. With the widespread application of computer technology in the machinery industry, intelligent algorithms and artificial intelligence technologies have been widely used in manufacturing systems, making it possible to obtain, represent, transmit, and store manufacturing information and knowledge, and to reason about it. Consequently, a new production model of intelligent manufacturing has emerged [5-6]. Aerospace reducers have the functions of changing transmission direction, speed regulation and torque matching, and are an indispensable part of the transmission system. Aerospace reducers are equipped on aircraft engines, rudders and other components. The research presented in this paper is motivated by the actual needs of manufacturing enterprises. After investigating enterprises, it was found that the processing of aerospace reducers involves multiple precise and complex links, which contain a huge amount of information. Reasonable arrangement of tasks, resources and time in the processing of aerospace reducers can improve processing efficiency and reduce processing costs. The aerospace reducer manufacturing sector presents an urgent need to leverage manufacturing process data, realize real-time analysis through correlation analysis and modeling of manufacturing service data, and reveal the adaptive mechanism of intelligent manufacturing services in an all-round and in-depth manner, so as to realize the quantification, optimization and intelligence of manufacturing services. Therefore, performing data processing and scheduling optimization on aerospace reducer processing operations is crucial. This study holds significant research value and practical importance.

The digitalisation of the workplace environment has progressed rapidly [7], In the field of grid and cloud computing, Casanova et al [8] proposed an adaptive scheduling algorithm that can automatically perform dynamic resource selection and collaborative allocation, and proposed heuristic algorithms such as Min-min, Max-min and Sufferage for scheduling independent tasks. Zhang et al [9] proposed an improved genetic algorithm to solve a class of flexible job shop scheduling problems with transportation time, and proved that the algorithm can effectively solve the problem by comparing with other algorithms. Aggarwal et al [10] proposed a general adaptive scheduling algorithm that can map task groups according to resource characteristics and task constraint relationships, with the goal of minimizing task execution time. Ruhu et al [11] proposed a genetic algorithm with three different constraint penalty functions to solve the simultaneous optimization of raw material procurement and production batches in batch production enterprises. Guettier et al [12] proposed a constraint model method using flow models to concurrently process production planning and scheduling. Nourmohammadi et al [13] designed a multi-objective colonial competition algorithm to solve the U-shaped assembly line balancing problem and obtained a better solution than the genetic algorithm. Yuan et al [14] used the delayed acceptance hill climbing algorithm to solve the bilateral assembly line type I problem with multiple constraints (regional constraints, location constraints and synchronization constraints) and obtained a better solution. In order to solve the batch flow integrated scheduling problem of a flow shop with the maximum completion time target, Professor Quanke Pan of Huazhong University of Science and Technology [15] proposed a discrete invasive weed optimization algorithm. Armes et al [16] looked forward to the application of big data technology in the field of aerospace manufacturing and discussed the focus and key technologies of big data applications in aerospace manufacturing. The above research provides a good research foundation for adaptive optimization scheduling of manufacturing services based on big data, and is an important guarantee for the success of this project.

In summary, advances in data processing and job scheduling optimization have made it feasible to enhance the quality of manufacturing processes. At present, the research on the FJSP problem mainly focuses on heuristic algorithms (such as genetic algorithms, particle swarm algorithms, etc.). Although heuristic algorithms can obtain high-quality approximate optimal solutions, their randomness, parameter sensitivity and long calculation time limit their application in actual production environments that require rapid response and frequent rescheduling of plans. In small-batch, multi-process, and high-value production scenarios, such as those for aerospace bevel gear shafts, there is a demand for a fast, stable, and deterministic scheduling method so that they can quickly generate a reliable scheduling plan that is better than manual experience when facing production disturbances. Therefore, the dynamic programming algorithm based on state transfer proposed in this paper offers a valuable approach for solving the flexible job shop scheduling problem.

2. Machining scheduling of bevel gear shafts based on dynamic programming algorithm

The flexible job shop scheduling problem is a complex combinatorial optimization problem, and dynamic programming algorithms are widely used in this field [17]. In the machining process of aerospace reducers, the machining of shaft components is more critical, with high precision requirements and long machining time, which has a huge impact on the production of the entire aerospace reducer. At the same time, the machining of shaft components requires more steps and links, and the human factors are relatively large. Therefore, this paper analyzes the machining operation scheduling of aerospace reducers by studying the operation link of bevel gear shaft machining.

2.1 Modeling of the processing job scheduling problem

The shaft system component is one of the main components of the aerospace reducer. The quality factors in its manufacturing process affect the quality prediction of the entire aerospace reducer. Therefore, this paper studies the corresponding manufacturing operation scheduling model based on the actual processing of the bevel gear shaft in the aerospace reducer through the dynamic programming algorithm; defines the manufacturing operation optimization target and quantifies it; analyzes the manufacturing performance constraints involved in the optimization, including comprehensive consideration of delivery time, operation time, etc., constructs a scheduling optimization model for the processing problem, and uses the dynamic programming algorithm for optimization and solution to solve the scheduling problems existing in the processing and manufacturing process.

Since the production workshop of the enterprise often has two different types of order requirements, the order requirements for processing operation scheduling are batch processing of bevel gear shafts of the same model, and the dynamic programming algorithm is just suitable for the case of overlapping sub-problems. Problems that can be solved by dynamic programming generally have the following three properties:

(1) Optimization principle: If the optimal solution of the problem contains the optimal solution of the sub-problem, then the problem is considered to have an optimized substructure, that is, it satisfies the optimization principle.

(2) No aftereffect: Once the state of a certain stage is determined, the decision after that state will not affect it. That is to say, the entire process after a certain state will not affect the previous state, but only related to the current state.

(3) Overlapping subproblems: The subproblems are not independent of each other, and the current subproblem may be used by the decision-making of the next stage.

When an enterprise processes a bevel gear shaft, the following assumptions were made for the workshop scheduling problem [18]:

(1) The processing operation must be completed after the previous operation is completed before the next operation can be carried out. The same workpiece can only be processed on a single machine.

(2) When the machine is in operation, the process that is already being processed cannot be interrupted. Only after the process is completed can the machine proceed to the next process.

(3) The processing technology and processing time of the workpiece are determined and will not change due to the processing sequence. There is no processing sequence constraint for different workpieces, and there is no priority for all processed workpieces.

(4) The initial state of all equipment is known, and all workpieces can be processed. Each equipment can only process one process at a time.

The symbolic description of the mathematical model for processing job scheduling is shown in Table 1:

Table 1

Symbols of the mathematical model for processing job scheduling

Variable	Meaning
N	Number of workpieces
P	Represents the set of all artifacts, $P = \{P_1, P_2 \dots P_n\}$
M	Represents the set of all machines, $M = \{M_1, M_2 \dots M_m\}$
Z	Delivery
$S_{M_{ij}^{(k)}}$	The start time of the j^{th} operation of machining the i^{th} workpiece using the k^{th} machine
$E_{M_{ij}^{(k)}}$	The end time of the j^{th} operation of the i^{th} workpiece using the k^{th} machine
O_{ij}	The j^{th} process of the i^{th} workpiece

$t_{M_{ij}^{(k)}}$	The time required to process the j^{th} step of the i^{th} workpiece using the k^{th} machine

Modeling of job shop scheduling: Assume that N workpieces of the same model $P_1 P_2 \dots P_n$ are assigned to M machines $M_1 M_2 \dots M_m$ for processing. Each workpiece needs to go through v processes $O_1 O_2 \dots O_v$. In order to improve production efficiency, the company provides three different processing machines for each process. Due to the performance differences between the machines, the time it takes for the three machines to process the same workpiece J in each process is different. However, the process and process of the workpiece are designed in advance and cannot be modified. This paper took the minimum completion time of all processes as the objective function, and achieves the requirements of the optimization scheduling goal by reasonably arranging the order of workpiece processing and the selection of machines.

The objective function of the scheduling model, as shown in Equation (1), is to minimize the maximum completion time among all workpieces. This is formulated as:

$$Min \max_{1 \le P \le n} \left(\sum_{j=1}^{v} t_{M_{ij}^{(k)}} \right) \tag{1}$$

The corresponding constraints are:

$$\begin{aligned} & S_{M_{ij}^{(k)}} < S_{M_{i(j+1)}^{(k)}} \qquad i = 1,2 \dots, n; j = 1,2 \dots, v; k = 1,2 \dots, m \\ & S_{M_{ij}^{(k)}} < E_{M_{ij}^{(k)}} \qquad i = 1,2 \dots, n; j = 1,2 \dots, v; k = 1,2 \dots, m \\ & \sum_{j=1}^{v} t_{M_{ij}^{(k)}} < Z_i \\ & S_{M_{ij}^{(k)}} > 0 \end{aligned} \tag{2}$$

$S_{M_{ij}^{(k)}} < S_{M_{i(j+1)}^{(k)}}$ ——The start time of the j^{th} process is before the start time of the $j+1^{\text{th}}$ process;

$S_{M_{ij}^{(k)}} < E_{M_{ij}^{(k)}}$ ——The start time of the j^{th} process of the i^{th} workpiece should be less than the end time;

$\sum_{j=1}^{v} t_{M_{ij}^{(k)}} < Z_i$——The completion time of all processes of the i^{th} job should be less than its delivery date

$S_{M_{ij}^{(k)}} > 0$——The start time of the j^{th} process of the i^{th} workpiece should be greater than 0.

2.2 Dynamic Programming Algorithm

2.2.1 Basic concepts and basic steps

The dynamic programming algorithm is based on Bellman's optimal control theory. It is suitable for discretizing a continuous nonlinear problem into a series of separate sub-problems and finding the optimal solution for each sub-problem at the lowest cost [19]. The principle of the dynamic programming algorithm is to solve the original problem by combining the solutions of the sub-problems. Generally speaking, the dynamic programming algorithm is mostly used in the case of overlapping sub-problems. After each sub-problem is solved, it is saved in a table. There is no need to repeatedly solve the sub-problem each time, which reduces the amount of calculation. Therefore, it is often used to deal with the job shop scheduling problem of the same batch model.

Its basic steps are as follows:

$Step1$: Define the original problem and sub-problems. The original problem is the final objective function to be solved, and the sub-problem is an objective function that is smaller in scale than the original problem but similar in scale.

$Step2$: Define the state. Determine the state that changes when solving the sub-problem.

$Step3$: Determine the state transition equation. Determine the transition relationship during the state change process and express it with the transition equation.

$Step4$: Determine the dynamic programming array. Save the results obtained when dynamically solving the sub-problem in the dynamic programming array (DP array).

2.2.2 Dynamic Programming Based on State Transfer

Aiming at the batch processing problem of bevel gear shafts of the same model, a dynamic programming model based on state transfer is proposed to optimize the job shop scheduling. Since batch processing of bevel gear shafts is a repeated sub-problem, it is advisable to use a dynamic programming algorithm to solve it. At the same time, the state transfer equation can not only show whether the machine is running in the algorithm, but also based on these machine changes, better understand the entire dynamic programming process. The state transfer is whether the equipment is in operation. If the equipment is running, it is 1, otherwise it is 0. The main ideas of its design are:

(1) Determine the current processing state π_m, of each processing machine. If the current processing state $\pi_m = 1$, it means that the current machine is processing a workpiece. Otherwise, it means that the machine is idle. Therefore, the machine state matrix corresponding to the processing of N workpieces and V processes should be:

$$\pi_{(N\times 3V)} = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1(3V)} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2(3V)} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{N1} & \pi_{N2} & \cdots & \pi_{N(3V)} \end{bmatrix} \tag{3}$$

(2) According to the actual processing situation of the bevel gear shaft of the enterprise, each process corresponds to three different processing machines, and the corresponding processing time is T_{M1} , T_{M2} , T_{M3} . Let's assume that the relationship between the length of the processing time is $T_{M1} \le T_{M2} \le T_{M3}$. Therefore, when initializing the first workpiece, each process should correspond to the one with the shortest processing time among the three machines. Therefore, when the first workpiece is processed, the processing machines corresponding to each process are $\{T_{M_{11}^{(1)}}, T_{M_{12}^{(1)}}, \ldots, T_{M_{1v}^{(1)}}\}$.

(3) To complete the rest of the DP array, it is necessary to continuously solve the processing machines that should be selected for the remaining workpieces. There are many different complex situations, such as whether to wait for the machine with the highest processing efficiency to complete and continue to use the machine for processing, whether to choose not to wait for the machine with the highest processing efficiency and choose the machine with the second highest processing efficiency among the three processing machines for processing, and for another example, when the i^{th} workpiece uses the machine with the best processing efficiency to complete the j^{th} process, the i^{th} workpiece can also be processed by the machine with the highest processing efficiency. In order to make the model

simpler, the idea is changed to, for t_{ij}(the processing time of the j^{th} process of the i^{th} workpiece)

①Determine the processing efficiency of the three processing machines in the j^{th} process, denoted as $\beta_j^{(1)}, \beta_j^{(2)}, \beta_j^{(3)}$ (where $\beta_j^{(1)}$) represents the highest processing efficiency among the three machines in the j^{th} process), and the corresponding processing time is denoted as $T_{M_{\beta_j^{(1)}}}, T_{M_{\beta_j^{(2)}}}, T_{M_{\beta_j^{(3)}}}$.

②Determine the workpieces that have recently used $\beta_j^{(1)}, \beta_j^{(2)}, \beta_j^{(3)}$ in the previous $i-1$ workpieces, that is, determine the workpieces with a value of 1 in the $3(j-1)+1 \sim 3j$ columns in the $\pi_{(N\times 3V)}$ matrix, denoted as ind_1, ind_2 and ind_3, respectively, and the corresponding total processing time of the previous j processes is:

$$\begin{aligned} Sum_{ind\_1} &= \sum_j t_{M_{ind\_1j}} \\ Sum_{ind\_2} &= \sum_j t_{M_{ind\_2j}} \\ Sum_{ind\_3} &= \sum_j t_{M_{ind\_3j}} \end{aligned} \tag{4}$$

The total processing time of the first j processes of the i^{th} workpiece is:

$$Sum_i = \sum_{j-1} t_{M_{ij}} \tag{5}$$

③ Determine the processing machine for the j^{th} process of the i^{th} workpiece. The processing time corresponding to $\beta_j^{(1)}, \beta_j^{(2)}, \beta_j^{(3)}$ is:

$$t = \begin{bmatrix} Sum_{ind\_1} + T_{M_{\beta_j^{(1)}}} \\ Sum_{ind\_2} + T_{M_{\beta_j^{(2)}}} \\ Sum_{ind\_3} + T_{M_{\beta_j^{(3)}}} \end{bmatrix} \tag{6}$$

Find the minimum value in t. The processing machine corresponding to the minimum value is the processing machine of the j^{th} process of the i^{th} workpiece.

2.3 Machining process analysis based on dynamic programming algorithm

The production workshop of this enterprise generally includes several major processes in the process of processing spare parts, namely: inspection, rough turning, drilling, tapping, polishing, quenching, cleaning, etc. Taking the processing technology of bevel gear shaft as an example, starting from the purchase of the workshop gear blank, the semi-finished product is first inspected and rough turned according to the requirements of the process card; then it goes through a series of process flows such as drilling, fine turning, tapping, polishing, etc.; carburizing quenching and related quality hardness testing are carried out according to the corresponding gear process card; shot peening and milling of parts are carried out; and finally, they are put into storage. The main process of bevel gear shaft processing can be simplified as shown in Fig. 1.

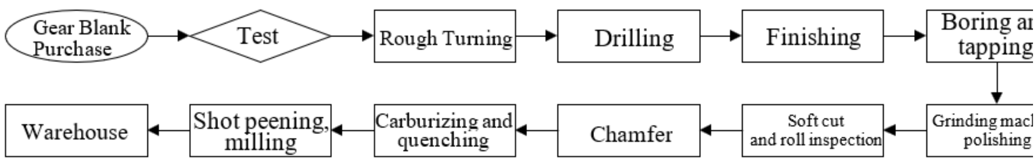

Fig. 1. Bevel gear shaft processing process

Processing requirements: The processing of the bevel gear shaft is mainly batch processing of the same model workpieces, which has the characteristics of typical workshop scheduling of manufacturing and processing enterprises. In addition, the production process of the spare parts is basically the same, so the dynamic programming algorithm based on state transfer can be used for scheduling and solving.

The process of processing the bevel gear shaft of this enterprise is simplified to 8 processes, each process has three processing machines, but there are differences in processing performance. The time used for each processing machine is shown in Table 2. Since the time difference of each process is not large, and in order to facilitate the Gantt chart, the processing time of each process is reduced by 1:5.

Table 2

Time required for each process of machining bevel gear shaft (min)

Workpiece	Rough Turning			Drilling			Finishing			Boring and tapping		
J1	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
	7	5	4	9	4	6	6	5	8	5	6	9
Workpiece	Grinding machine polishing			Chamfer			Carburizing and quenching			Milling		
J1	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24
	19	14	13	4	2	6	9	14	10	5	11	9

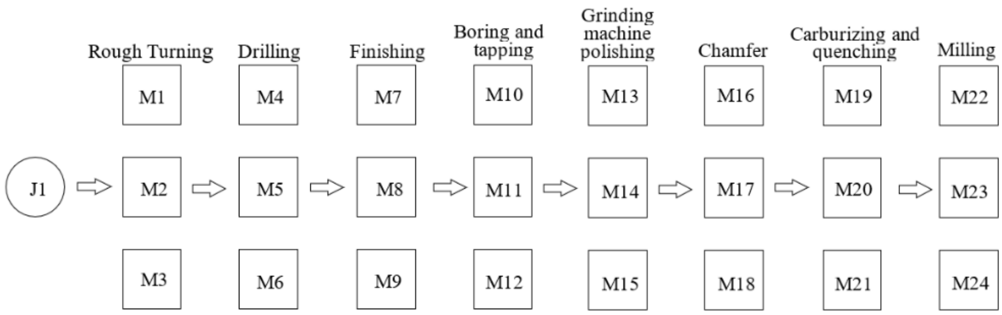

Fig. 2. Schematic diagram of the machine that can be selected for J1 workpiece in 8 processes

As shown in Fig. 2, there are 10 workpieces of the same model and batch being processed and produced. Assuming that the first step of the J1 workpiece is carried out on the M3 equipment, the J2 workpiece can only choose the M1 or M2 equipment for the first step, and cannot choose the M3 equipment. After the first step of the J1 workpiece is completed, the subsequent workpieces can be processed on the M3 equipment. The current processing flow of the enterprise **is** mainly based on machine availability. Under this manual scheduling policy, the time to complete 10 workpieces was approximately 110 minutes.

3 Solution results and analysis

3.1 Results

The dynamic programming algorithm was programmed and implemented using MATLAB R2023b. Then the time required for each process of bevel gear shaft processing was written into the program. After optimization by the dynamic programming model based on state transfer, the processing time of each process of each workpiece and the selected processing machine were shown in Table 3 and Table 4 respectively:

Table 3

Total processing timetable for each workpiece and each process (min)

Process / Workpiece	O1	O2	O3	O4	O5	O6	O7	O8	Time
J1	4	4	5	5	13	2	9	5	47
J2	5	6	6	6	14	2	10	5	54
J3	7	5	6	5	19	2	9	6	59
J4	8	8	7	5	16	2	13	5	64
J5	10	7	6	6	22	2	9	7	69
J6	14	6	8	5	24	2	10	5	74
J7	12	9	8	6	26	2	9	7	79
J8	15	8	8	7	27	2	12	5	84
J9	21	4	8	8	29	2	9	8	89
J10	16	13	6	8	36	2	9	5	95

Table 4

Table of selected processing machines for each workpiece and each process

Process / Workpiece	O1	O2	O3	O4	O5	O6	O7	O8	Time
J1	M3	M5	M8	M10	M15	M17	M19	M22	47
J2	M2	M6	M7	M11	M14	M17	M21	M22	54
J3	M1	M5	M8	M10	M13	M17	M19	M22	59
J4	M3	M5	M8	M10	M15	M17	M21	M22	64
J5	M2	M6	M7	M11	M14	M17	M19	M22	69
J6	M1	M5	M8	M10	M15	M17	M21	M22	74
J7	M3	M4	M7	M11	M13	M17	M19	M22	79
J8	M2	M6	M9	M10	M14	M17	M21	M22	84
J9	M1	M5	M8	M11	M15	M17	M19	M22	89
J10	M3	M5	M7	M10	M14	M17	M19	M22	95

The J1 workpiece is processed in the shortest time of each process and each equipment, that is, it is processed by 8 equipment M3, M5, M8, M10, M15, M17, M19, and M22, and the time used is 47 minutes. When the first step of the J1 workpiece is processed by the M3 equipment, the first step of the J2 workpiece can only be processed by the M2 equipment with less time. After the first step of the J1 workpiece is completed, the second step of the J2 workpiece is processed by the M5 equipment after 1 unit. The first step of the J2 workpiece is completed because the M5 equipment is running. The J2 workpiece has two options, one is to use the M5 equipment after waiting for 3 units, and the other is to use the M6 equipment for the second step. The first option requires 7 units to complete the second step, while the second option only requires 6 units, so the M6 equipment is selected for processing, and so on. The equipment waiting time is also counted into the processing time. After optimization by the dynamic programming model based on state transfer, the Gantt chart of the processing scheduling of each workpiece process is obtained as shown in Fig. 3.

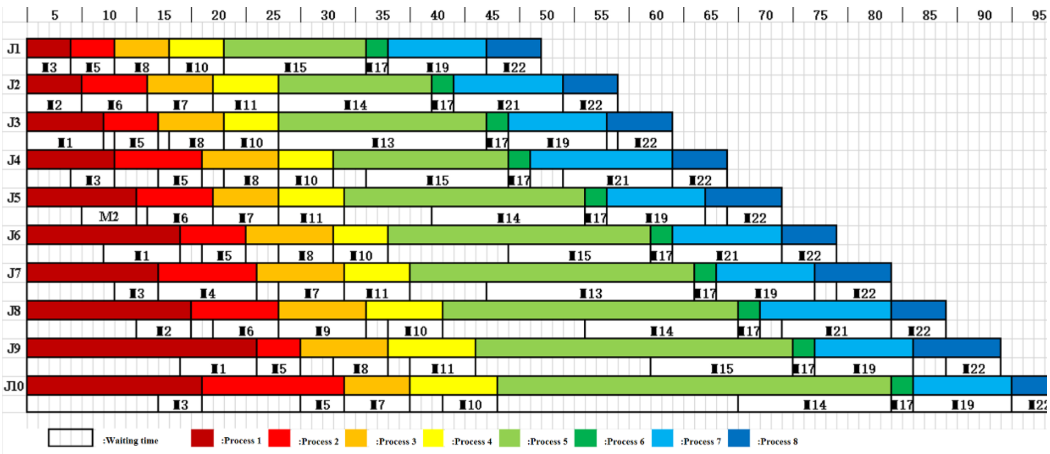

Fig. 3. Gantt chart of production scheduling for each workpiece process

The iterative process was implemented in MATLAB R2023b. After the model was optimized, the equipment waiting time was shortened and the equipment was reasonably arranged for processing. The shortest time required to complete the processing of 8 processes of 10 workpieces in the same batch was 95 minutes. Compared with the original processing time of 110 minutes, the processing time was shortened by 13%.

Table 5

Operation status of each device

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24
J1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	1	0	0
J2	0	1	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	0
J3	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0
J4	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0
J5	0	1	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0
J6	1	0	0	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0
J7	0	0	1	1	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	1	0	0
J8	0	1	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	1	0	0
J9	1	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0
J10	0	0	1	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0

Through the dynamic programming algorithm of state transfer, it can be expressed that when the equipment is in operation, the value is 1, otherwise it is 0. It can be seen from Table 5, that the equipment M12, M16, M18, M20, M23 and M24 do not participate in the processing steps after optimization, therefore, these six machines do not participate in the optimized schedule. This suggests they could potentially be removed from this process flow, leading to reduced operational costs.

3.2. **Comparative analysis**

To further validate the proposed algorithm's effectiveness, its performance was benchmarked against the Genetic Algorithm (GA) widely used in the current flexible job shop scheduling problem. The genetic algorithm (GA) in this paper is implemented based on the PYTHON language. The main parameters of the algorithm are shown in Table 2.6. Its function is implemented through PYCHARM. After iteration, the shortest processing time is 102 minutes, which is 7% shorter than the original processing time of 110 minutes.

Table 6

Main parameters of genetic algorithm

Population Size	Maximum Generations	Crossover Rate	Mutation Rate	Selection Method	Tournament Size	Elitism Size
100	300	0.8	0.1	Tournament Selection	5	2

Table 7 presents the computation time and results of the two algorithms.

Table 7

Algorithm efficiency comparison

Dynamic Programming(DP)		Genetic Algorithms(GA)	
Shortest completion time(min)	Execution Time(s)	Shortest completion time(min)	Execution Time(s)
95	0.001	102	15.123

The following conclusions can be drawn from the data in the comparison table: When solving small and medium-scale scheduling problems common in the aerospace field, the dynamic programming algorithm based on state transfer proposed in this paper has significant advantages in terms of solution optimality and calculation speed, which proves the effectiveness of the algorithm in this paper.

4. Conclusions

This study aims at the scheduling optimization problem of bevel gear shafts of aerospace reducers in small batch and multi-process production mode, and proposes a scheduling optimization method based on state transfer dynamic programming algorithm. By constructing a mathematical model with the goal of minimizing the maximum completion time and solving it with the proposed algorithm, the scheduling optimization of the bevel gear shaft processing process is completed. Through the verification of actual enterprise cases and the performance comparison with traditional genetic algorithms, the main conclusions are as follows:

(1) Compared with the company's original scheduling plan, this method can shorten the maximum completion time from 110 minutes to 95 minutes, a reduction of 13%. At the same time, the optimization results also identified 6 redundant equipment that were not used in the entire processing process, providing direct data

support for the company to achieve equipment streamlining and cost control.

(2) Comparative analysis with genetic algorithms shows that the dynamic programming algorithm has significant advantages over traditional heuristic algorithms when solving small and medium-scale scheduling problems, which verifies the effectiveness of the method proposed in this paper.

(3) For the small- and medium-scale, high-value scheduling problems common in the aerospace field, deterministic optimization algorithms (such as dynamic programming) present a more competitive alternative to stochastic heuristic algorithms, providing a reference value for future research on flexible job shop scheduling problems.

It should be pointed out that the dynamic programming method proposed in this paper has limitations in scalability. The case mentioned in this paper is intended to verify the effectiveness of the algorithm on a small scale. However, dynamic programming has the problem of "dimensionality curse". For real production scenarios with dozens of different parts and different process routes, the computational cost of this algorithm becomes prohibitively high. Therefore, the practical application of this method is suitable for small and medium-scale scheduling problems commonly seen in the aerospace field. For larger and more complex scheduling problems, industry and academia usually use heuristic and meta-heuristic methods, such as genetic algorithms (GA), particle swarm optimization (PSO) or simulated annealing.

Although the assumptions made in this paper on the scheduling problem of the machining job shop are necessary conditions for building a mathematical model, they are also idealized treatments of the real manufacturing environment. Future research should focus on establishing a stochastic or robust optimization model that can handle the uncertainty of processing time, further narrowing the gap between theoretical models and the demands of industrial practice.

Acknowledgments

This work was supported in part by the National Defense Science and Technology Project Foundation under Grant 0106142, in part by the Ministry of Education in China Project of Humanities and Social Sciences under Grant 17YJC630139, and in part by the Fundamental Research Funds for the Central Universities under Grant 30917011303.

REFERENCES

[1] *C. Yang, S. Lan, W. Shen, G. Q. Huang, X. B. Wang, T. Y. Lin*, Towards product customization and personalization in IoT-enabled cloud manufacturing, cluster computing, Vol. **20**, Iss. 2, 2017, pp. 1717-1730.

[2] *Q. Y. Ma, Y. M. Zhao, T. Gong*, Research on production scheduling of remanufacturing enterprises under cloud manufacturing model, Combined machine tools and automated processing technology, DOI:10.13462/j.cnki.mmtamt.2024.09.038.

[3] *F. Kache, S. Seuring*, Challenges and opportunities of digital information at the intersection of big data analytics and supply chain management, International Journal of Operations & Production Management, Vol. **37**, Iss. 1, 2017, pp. 10-36.

[4] *J. K. Jee, G. H. Kim*, Potentiality of big data in the medical sector: focus on how to reshape the healthcare system, Healthcare informatics research, Vol. **19**, Iss. 2, 2013, pp. 79-85.

[5] *D. M. Lv, X. L. Sun, Z. Cheng*, Research on machining center spindle bearing fault diagnosis based on PARD - BP algorithm, U.P.B. Sci. Bull. Series D, Vol. **86**, Iss. 1, 2024.

[6] *D. X. Sun, W. Li, Y. Shen, H. Chen, Z. H. Tan, X. Ding*, Application practice of intelligent algorithm and digital twin in electromechanical equipment, U.P.B. Sci. Bull., Series D, Vol. **86**, Iss. 4, 2024.

[7] *E. Lazarou, C. Mustata, C. Dragomirescu*, Working and learning in industry 4.0 environments, U.P.B. Sci. Bull., Series D, Vol. **81**, Iss.4, 2019.

[8] *H. Casanova, G. Obertelli, F. Berman, R. Wolski*, The AppLeS Parameter Sweep Template: User-Level Middleware for the Grid, Sci. Program., Vol. **8**, Iss. 3, 2001.

[9] *G. H. Zhang, J. H. Sun*, Solving the flexible job shop scheduling using a new machine neighborhood structure, U.P.B. Sci. Bull., Series D, Vol. **82**, Iss. 3, 2020.

[10] *A. Aggarwal, R. D. Kent*, Resource Characteristic Based Optimization for Grid Scheduling, Proceedings of 2010 International Colloquium on Computing, Communication, Control, and Management (CCCM2010), Vol. **4**, 2010, pp. 143-146.

[11] *R. Sarker, C. Newton*, A genetic algorithm for solving economic lot size scheduling problem, Vol. **42**, Iss. 2-4, 2002, pp. 189-198.

[12] *C. Guettier, B. Allo, V. Legendre, et al*, Constraint Model-Based Planning and Scheduling with Multiple Resources and Complex Collaboration Schema// Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems, April 23-27, 2002, Toulouse, France. DBLP, 2002.

[13] *A. Nourmohammadi, M. Zandieh, R. Tavakkoli-Moghaddam*, An imperialist competitive algorithm for multi-objective U-type assembly line design, Journal of Computational ence, Vol. **4**, Iss. 5, 2013, pp. 393-400.

[14] *B. Yuan, C. Zhang, X. Shao*, A late acceptance hill-Climbing algorithm for balancing two-sided assembly lines with multiple constraints, Journal of Intelligent Manufacturing, Vol. **26**, Iss. 1, 2015, pp. 159-168.

[15] *H. Y. Sang, Q. K. Pan*, Discrete Invasive Weed Optimization Algorithm for Solving Batch Flow Integrated Scheduling in Flowshops, Control Theory and Applications, Vol. **32**, Iss. 2, 2015, pp. 393-400.

[16] *T. Armes, M. Refern*, Using Big Data and predictive machine learning in aerospace test environments, [IEEE 2013 IEEE AUTOTESTCON - Schaumburg, IL, USA (2013.09.16-2013.09.19)] 2013 IEEE AUTOTESTCON, 2013:1-5.

[17] *S. Q. Xiao*, Optimization of Krill Herd Algorithm and Its Application in Flexible Workshop Scheduling Problem, PhD Thesis, Changsha University of Science and Technology, 2019.

[18] *L. Zhang*, Research on Flexible Job Shop Scheduling Problem Based on Multi-objective Optimization Algorithm, PhD Thesis, Shandong Agricultural University, 2019.

[19] *S. X. Zhang, B. Wu*, Speed ratio optimization of electric car transmission system based on dynamic programming algorithm, Journal of Shaanxi University of Science and Technology (Natural Science Edition), Vol. **36**, Iss. 5, 2020, pp. 34-40+88.