**International Journal of Education and Social Development**
ISSN: 3078-9931 (Print), ISSN: 3078-2287 (Online) | Vol. 7, No. 3, 2026

# Research on Flexible Job Shop Scheduling Optimization Based on Improved Genetic Algorithm

**Junlan Feng**

School of Economics and Management, Southwest Petroleum University, Chengdu, China

**Abstract:** Aiming at the engineering pain points of difficult solving in practical production, slow convergence of traditional genetic algorithms, and easy getting stuck in local optima, an improved genetic algorithm with the goal of minimizing the maximum completion time has been designed. By improving coding, crossover, and introducing roulette wheel selection methods, the overall algorithm's global optimization capability is enhanced. The comparison of experimental results shows that the improved genetic algorithm is superior to the traditional genetic algorithm in optimizing the target solution.

**Keywords:** Improved genetic algorithm; Flexible workshop scheduling; automobile production.

## 1. Preface

With the progress of the times, the global manufacturing industry is developing towards intelligence and digitization. At present, China is in the initial stage of the 15th Five Year Plan, which is also a critical period. The country has clearly proposed to accelerate the development of new quality productive forces, promote the high-end, intelligent, and green transformation of the manufacturing industry. Combining digitalization with lean manufacturing tools has become a key strategy for improving operational performance in complex production environments [1]. As the core field of high-end equipment manufacturing, automobile manufacturing has become the mainstream of customized orders with multiple varieties, small batches, and short delivery times. Improving efficiency has become a goal that many enterprises have been optimizing. Job scheduling, that is, how to arrange production, has become the key to improving efficiency.

Flexible Job Shop Scheduling Problem (FJSP) is an extension of Job Shop Scheduling Problem (JSP) that adds flexibility to machine processing. It breaks the limitation that a single device can only correspond to a single process, and allows processes to be processed on multiple machines [2].By optimizing the sequence of processes and machine allocation, equipment load can be effectively balanced, completion time can be shortened, and overall workshop efficiency can be improved. The research on workshop scheduling problems has lasted for decades, and Johnson studied the flow shop scheduling problem as early as 1954 [3]; Arthanari also solved the mixed flow shop scheduling problem using the branch and bound method as early as 1971, using a two-stage scheduling model [4]; In 1984, some scholars also mentioned the workshop scheduling problem in their own research [5]. In general, early research mainly used mathematical programming and heuristic rule methods as the main solving methods to solve simple scale flexible scheduling problems by constructing integer programming models, mixed integer programming models, etc. However, these methods have the drawbacks of low solving efficiency and difficulty in adapting to complex scenarios, and cannot meet the practical production needs of large-scale and multi constrained situations.

Nowadays, relevant research is becoming increasingly mature, and there are more and more studies on FJSP. However, in practical scenarios such as automotive parts with complex working conditions, strong process coupling, and high equipment flexibility, existing algorithms still have some shortcomings. Therefore, this article combines the actual working conditions of automotive parts production and designs an improved genetic algorithm with stronger adaptability, higher stability, and better solution accuracy to address the shortcomings of existing algorithms.

## 2. Problem Description and Mathematical Modeling

### 2.1. Problem description

Company A is a factory specializing in the production of automotive parts, with a processing flow that conforms to the description of flexible job shop scheduling problems. There are a total of n types of components to be processed in its processing workshop, grouped together as $J = \{J_1, J_2, \cdots, J_i, \cdots, J_n\}$, $i = 1,2,\cdots n$; The collection of m processing machines and equipment is $M = \{M_1, M_2, \cdots M_k, \cdots, M_m\}$, $k = 1,2,\cdots m$; The process set is $O = \{O_{i1}, O_{i2}, \cdots, O_{ij}\}$,

Disregarding factors such as equipment failures, urgent cutting of workpiece orders, and transportation time during the production process. In the actual production process, relying solely on the experience of employees to arrange the processing sequence may result in long equipment waiting times, extended production cycles, reduced equipment utilization, and increased production costs. Therefore, this study aims to scientifically arrange the processing sequence of workpieces in a reasonable manner, so as to minimize the total time required for all workpieces to be completed, and improve the overall production efficiency and equipment utilization of the processing workshop as the scheduling optimization objective.

Based on the characteristics of the standard genetic algorithm research in this study, the following assumptions must be met:

(1) Each workpiece can only be processed on one device at a time;

(2) The same process of the same workpiece can be

selected from multiple machines for processing;

(3) Each device can only process one workpiece at a time;

(4) Each workpiece has different processing times on different equipment;

(4) The process route for each workpiece is completely consistent, with a fixed and unchangeable sequence of operations;

(6) Once each workpiece starts processing, interruption is not allowed;

(7) Ignoring interference factors such as transportation time between processes, equipment failures, personnel changes, and replacement of equipment components, the research scenario is deterministic scheduling;

(8) Without considering other optimization objectives such as processing costs, energy consumption, equipment overload, etc., the optimization index is only based on minimizing the maximum completion time.

The relevant parameters for establishing the model are shown in Table 1:

**Table 1.** Related symbols and descriptions

| Symbol | Descriptions |
|---|---|
| $i$ | Workpiece Number |
| $k$ | Machine equipment number |
| $n$ | Total number of workpieces |
| $m$ | Total number of machinery and equipment |
| $j$ | Process Sequence Number |
| $h_i$ | Total number of processes for workpiece i |
| $O_{ij}$ | The $j_{th}$ process of workpiece i |
| $M_{ij}$ | The set of machines capable of processing the $j_{th}$ process of workpiece i |
| $t_{ijk}$ | The time for the $j_{th}$ process of workpiece i to be processed on machine k |
| $S_{ijk}$ | The time when the $j_{th}$ process of workpiece i starts processing on machine k |
| $C_{ijk}$ | The time it takes for the $j_{th}$ process of workpiece i to be completed on machine k |

## 2.2. Mathematical model

### 2.2.1. Minimize maximum completion time

Find the latest completion time among all workpieces, processes, and machines, which is the total time it takes for all workpieces to be processed, and then take the minimum value. To achieve the highest production efficiency with the minimum total completion time as the goal, the expression is:

$$\min\left(C_{\max}\right) = \min\left(\max(C_{ijk})\right) \tag{2-1}$$

### 2.2.2. Constraints

All models must comply with the following constraints:

$$C_{ijk} = S_{ijk} + t_{ijk}, i \in J, j \in O, k \in M \tag{2-2}$$

$$S_{i(j+1)k'} \geq C_{ijk}, i \in J, j \in O, k \in M \tag{2-3}$$

$$S_{ijk} \geq C_{i'j'k}, i, i' \in J, j, j' \in O, k \in M \tag{2-4}$$

$$\sum_{k=1}^{m} x_{ijk} = 1, i \in J, j \in O, k \in M \tag{2-5}$$

$$C_{ih_ik} \leq C_{\max}, i \in J, k \in M \tag{2-6}$$

$$S_{ijk} \geq 0, C_{ijk} \geq 0, i \in J, j \in O, k \in M \tag{2-7}$$

Equation (2-2) represents that the completion time of a process is equal to the start time plus the processing time; Equation (2-3) indicates that the same workpiece must complete the previous process before starting the next process; Equation (2-4) indicates that the same machine can only perform one process at a time; Equation (2-5) indicates that each process must and can only select one machine for processing; Equation (2-6) indicates that the completion time of each workpiece cannot exceed the total completion time; Equation (2-7) indicates that both the start time and completion time are non negative.

# 3. Design and Application of Improved Genetic Algorithm

## 3.1. Improved genetic algorithm

Genetic algorithm (SAG) has strong adaptability and flexibility since its inception, and can effectively explore and find the optimal or near optimal solution. However, standard genetic algorithm has problems such as slow convergence speed in practical applications, which reduces the efficiency of problem-solving and wastes resources [6]. Therefore, this study will conduct optimization research by improving the genetic algorithm, and the algorithm process is shown in Figure 1.

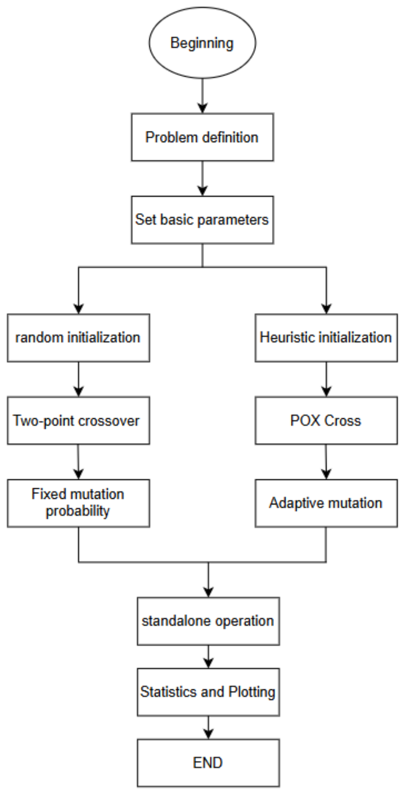


**Figure 1.** Improve the flowchart of genetic algorithm

## 3.2. Initial population generation

The initial population generation is the first step of genetic algorithm, and it is necessary to construct the initial population before the algorithm begins optimization. This article will prioritize the selection of equipment that can minimize the current process waiting time and the maximum completion time based on the priority of workpieces and processes, as well as the availability of machine equipment, for reasonable allocation. The specific steps are as follows: (1) Set the population size and determine the number of scheduling plans that need to be generated; (2) Randomly generate process layer codes for each scheme, and then assign a machine to each process to form a machine layer code; (3) Merge the process layer encoding and machine layer encoding to generate a complete two-layer encoding.

## 3.3. Encoding and Decoding

For the flexible job shop problem (FJSP), the encoding adopts a dual layer encoding MSOS approach. MS is the machine coding part, which represents the sequential number of the processing machine selected for the current process in the available machine set; OS is the process coding part. After the processing machine of the process is determined, a process based coding method is adopted, and the chromosome length of MS and OS is equal to the sum of all processes. As shown in Figure 2, the OS layer represents the workpiece serial number, and the number of occurrences represents the process serial number. The workpieces are numbered in order and then assigned codes through random permutation, which can clearly reflect the processing priority of the workpieces; The MS layer corresponds to the equipment number of the processing machine.

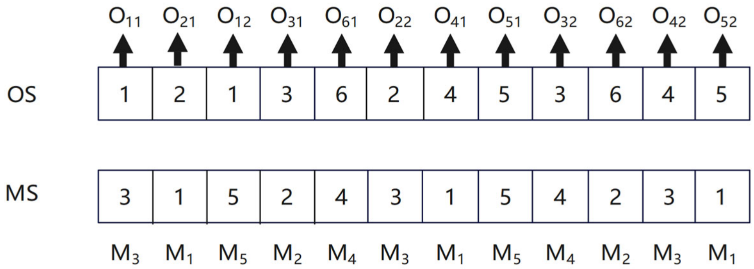


**Figure 2.** MSOS Double Layer Encoding Diagram

## 3.4. Fitness function

The fitness function is usually used to evaluate the quality of scheduling schemes. Chromosomes with higher fitness are more likely to be retained for subsequent operations during the selection process. In this study, with the goal of minimizing the maximum completion time, the fitness function is as follows:

$$Fit = \frac{1}{C_{\max}} \tag{3-1}$$

## 3.5. Selection

The selection operation corresponds to the "survival of the fittest" in the origin of species. In this study, the roulette wheel selection method was adopted, as shown in formula (3-2), which is an innovative approach aimed at optimizing the selection process in genetic algorithms. This algorithm operates on a given list of chromosome fitness scores, with the core goal of selecting suitable chromosomes based on adaptive probabilities [7]. The operation method is to calculate the probability of chromosome selection, then place it on a disk and present a fan-shaped statistical chart. Then, the algorithm rotates the disk and randomly lands on each part, indicating which chromosome to choose.

$$p_i = \frac{Fit_i}{\sum_{i=1}^{N} Fit_i} \tag{3-2}$$

In the equation, Pi represents the probability of selection. The greater the fitness, the higher the probability of fitness, and the larger the corresponding area on the wheel, the higher the probability of being selected.

## 3.6. Intersection

Crossover is the core operation in genetic algorithms to generate new individuals and enrich population diversity. Mainly by recombining the excellent fragments of the parent generation onto the offspring individuals, the fitness of the population is improved. Based on the characteristics of the homework problem in this study, the process coding part will choose POX crossover, and the crossover process is shown in Figure 3:

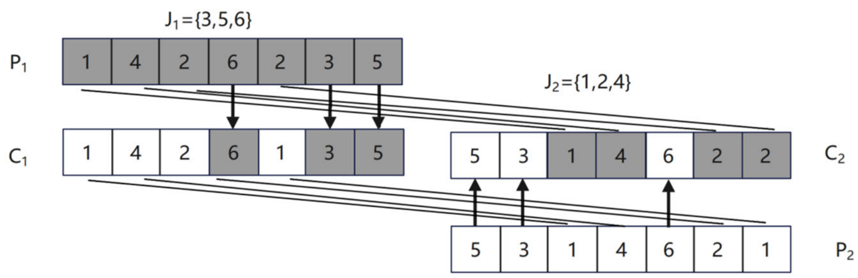


**Figure 3.** POX Cross plot

(1) Randomly select two parent bodies, $P_1$ and $P_2$, from the current population and extract their process layer codes;

(2) Randomly divide all workpieces into two non empty subsets, $J_1$ and $J_2$ , where$J_1 \cup J_2$ is the set of all workpieces;

(3) Generate offspring $C_1$: First, copy the workpiece numbers belonging to subset $J_1$ from parent $P_1$ directly to $C_1$ according to their original positions; Fill in the workpiece numbers belonging to subset $J_2$ in $P_2$ in the original order to the vacant positions of sub generation $C_1$, and the complete sub generation $C_1$ can be obtained after the operation is completed;

(4) Generate offspring $C_2$: First, copy the workpiece numbers belonging to subset $J_1$ from parent $P_2$ directly to $C_2$ according to their original positions; Fill in the workpiece numbers belonging to subset $J_2$ in $P_1$ in the original order to the vacant positions of offspring $C_2$, and complete the operation to obtain the complete offspring $C_2$;

(5) $C_1$ and $C_2$ are the offspring obtained after crossing.

The machine layer encoding adopts two-point crossing, as shown in Figure 4:

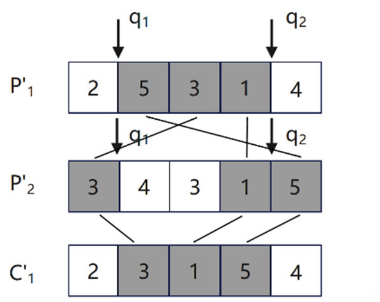


**Figure 4.** Schematic diagram of two-point intersection

(1) Randomly generate two crossing points $q_1$ and $q_2$ on two parental chromosomes $P'_1$and $P'_2$, and divide the chromosome into three parts;

(2) Select the middle part number of parent $P'_1$, search for the number in parent $P'_2$, and replace the middle part of $P'_1$ directly without changing the order;

(3) After replacement, the offspring chromosome $C_1$'is formed.

## 3.7. Mutation

The purpose of mutation operation is to maintain the diversity of the population and avoid the trap of the algorithm falling into local optima in advance. Mutation can find a globally optimal scheduling scheme by randomly modifying the encoding of individuals with a lower probability. This study will use exchange mutation to perform mutation operations, randomly swapping two positions while ensuring that the number of codes in each segment remains unchanged. For example, as shown in the figure, the upper layer is the parent and the lower layer is the new offspring. After completing the exchange, it is necessary to check the number of occurrences of each code in the offspring to ensure consistency with the parent, without duplication or omission.

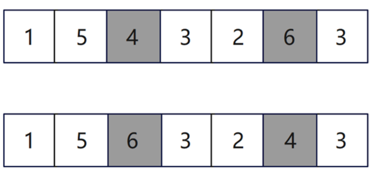


**Figure 5.** Variation diagram

# 4. Simulation Results and Analysis

## 4.1. Experimental data foundation and parameter settings

### 4.1.1. Experimental Data Fundamentals

To verify the effectiveness and superiority of the Improved Genetic Algorithm (IGA) in solving the Flexible Job Shop Scheduling Problem (FJSP), simulation experiments were conducted with the goal of minimizing the maximum completion time. The data used in this article comes from a company specializing in the production of automotive parts. Its processing workshop mainly involves six types of basic parts, and each workpiece has different numbers of processes. The processes and processing time are shown in Table 2.

**Table 2.** Initial Processing Data Sheet

| Workpiece | Process | Machine processing time (min) | | | | |
|---|---|---|---|---|---|---|
| | | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ |
| $J_1$ | $O_{11}$ | 10 | - | - | - | - |
| | $O_{12}$ | - | 5 | - | - | 19 |
| $J_2$ | $O_{21}$ | 19 | 19 | - | 5 | 6 |
| | $O_{22}$ | - | - | - | - | 6 |
| | $O_{23}$ | - | - | 17 | 5 | 20 |
| $J_3$ | $O_{31}$ | 15 | 8 | - | 9 | 5 |
| $J_4$ | $O_{41}$ | 11 | 16 | - | - | 11 |
| | $O_{42}$ | - | - | 15 | - | - |
| $J_5$ | $O_{51}$ | - | - | 7 | 11 | 19 |
| $J_6$ | $O_{61}$ | - | - | 17 | - | 10 |
| | $O_{62}$ | - | - | 14 | - | 6 |
| | $O_{63}$ | 5 | 19 | 14 | 10 | 14 |

### 4.1.2. Environment and parameter settings

The simulation experiment platform is the MATLAB R2023a simulation software, using a computer processor Intel (R) Core (TM) i5-1035G1 CPU @ 1.00GHz (1.19 GHz), with a built-in RAM of 16.0GB, and a system version of Windows 11. The parameter settings are shown in Table 3. To ensure the fairness of the comparative experiment, the traditional genetic algorithm and the improved genetic algorithm use completely identical parameter settings, with only some differences in certain operations

**Table 3.** Parameter Example Table

| Population size | Maximum number of iterations | Cross frequency | Variation frequency |
|---|---|---|---|
| 70 | 100 | 0.85 | 0.05 |

## 4.2. Simulation result analysis

The figure 6 shows the convergence trends of two algorithms in typical runs. From the graph, it can be clearly observed that the decreasing slope of the dashed line curve (IGA) is significantly greater than that of the solid line curve (GA) at the beginning of the iteration. This indicates that the improved algorithm can quickly lock in high-quality solution areas in the early search stage by optimizing the initial population generation strategy. In the later stages of iteration, the curve of traditional GA tends to flatten, and the maximum completion time stagnates between 33-35, indicating that the algorithm has fallen into a local optimum situation. In contrast, the IGA curve continues to exhibit a fluctuating downward trend and eventually converges to a lower numerical range. From the overall trend of the convergence curve, the iteration curve of the Improved Algorithm (IGA) is always below that of the Traditional Algorithm (GA), indicating that at each stage of evolution, the Improved Algorithm can maintain higher quality population fitness, thereby achieving shorter completion time at the final convergence. This verifies that the improvement strategy introduced in this article effectively enhances the algorithm's ability to jump out of local optima. Although the convergence algebraic mean is slightly higher than that of traditional GA, this small computational cost is exchanged for higher solution quality.

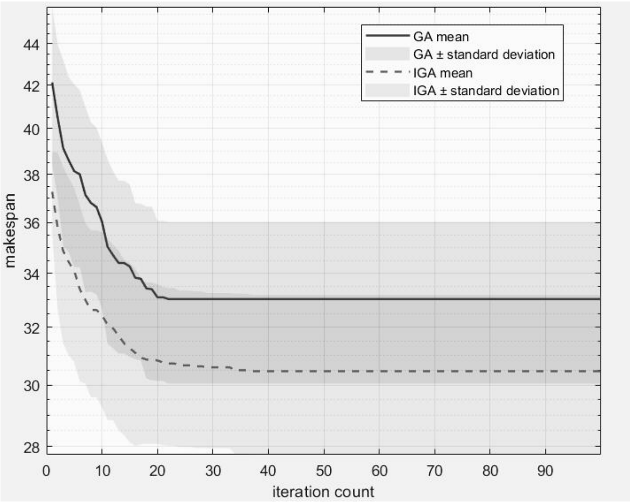


**Figure 6.** Comparison chart of convergence curves

In order to comprehensively evaluate the robustness of the algorithm, performance indicators of two algorithms with 30 iterations were calculated. As shown in the table, the average optimal completion time of IGA is 30.47 minutes, which is 7.77% higher than the traditional GA's 33.03 minutes. However, in terms of convergence speed and stability, the experimental results exhibit more complex characteristics. Although the improved algorithm performs well in terms of the quality of the final solution, its average convergence algebra is basically the same as traditional algorithms, and it does not show significant acceleration effect, and even converges slightly slower in some runs.

**Table 4.** Iteration Results Table

| Indicator | Makespan (min) | | Convergent algebra | |
|---|---|---|---|---|
| | Mean | Standard deviation | Mean | Standard deviation |
| GA | 33.03 | 2.99 | 25.03 | 6.36 |
| IGA | 30.47 | 2.73 | 25.53 | 9.05 |
| Improvement range | +7.77% | | - | |

In addition, by observing the box plot in Figure 7, it can be found that although the overall box of the improved genetic algorithm moves downwards, there are still long whiskers and outliers above it, and the standard deviation of the convergence algebra is higher than that of the traditional algorithm. This indicates that although the improved algorithm is likely to find better solutions, in rare cases there may still be fluctuations in the results, showing a certain degree of instability. This may be related to the algorithm sacrificing some local development certainty when enhancing global search capabilities. That is to say, although the improved genetic algorithm has a high probability of finding high-quality solutions, it still faces the risk of "premature convergence" in rare cases, resulting in significant fluctuations in the results.

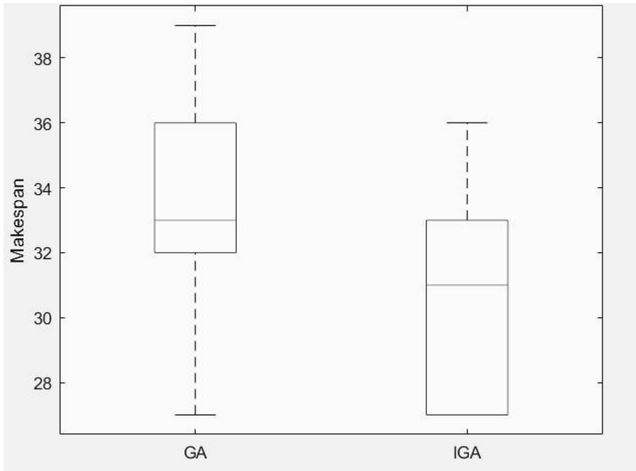


**Figure 7.** Box plot of optimal value distribution

Figure 8 shows the global optimal scheduling scheme found by IGA during 30 independent runs. From the overall topology of the Gantt chart, the scheduling scheme exhibits high compactness and continuity, verifying the algorithm's optimization ability under complex constraint conditions. Specifically, the horizontal axis in the figure represents the time dimension, the vertical axis represents machine resources, and different colored rectangular blocks represent specific processes of different workpieces. Observing the operating trajectories of M1 and M5, it can be observed that from time 0 until the end of the 27th minute, the load color blocks of these two key devices almost completely filled the entire timeline, with no obvious idle time segments in between. This indicates that the algorithm has successfully achieved "zero waiting" scheduling on the critical path, maximizing the capacity of bottleneck devices. Meanwhile, for machine M4, although there are a small amount of machining gaps, these gaps are reasonably arranged within the waiting period of non critical processes, without causing any drag on the overall completion time. In addition, from the perspective of workpiece circulation, the switching and connection between different processes of the same workpiece on different machines are closely connected, effectively reducing the waiting time of workpieces in the queue between processes and avoiding cycle extension caused by logistics stagnation. In summary, the visualization results not only demonstrate the feasibility of the obtained solution in mathematical logic, but also demonstrate the superiority of this scheme in balancing machine loads and eliminating invalid waiting from an engineering practice perspective.

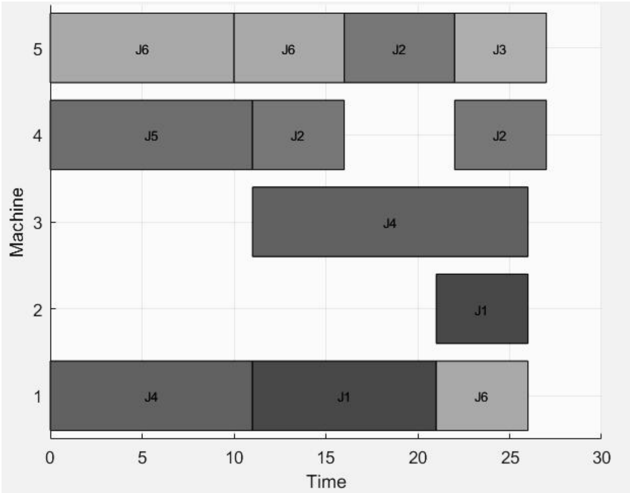

**Figure 8.** IGA Optimal Scheduling Gantt Chart

# 5. Conclusion

This article establishes a flexible job shop scheduling model with the goal of minimizing the maximum completion time, and proposes an improved genetic algorithm for solving it.

(1) The initialization of the population introduces heuristic rules, which not only preserves the diversity of the population, but also significantly increases the proportion of high-quality individuals in the initial population, making the algorithm in a better solution space at the initial stage, thereby accelerating the convergence speed;

(2) Based on the process layer, the improved genetic algorithm selects POX crossover, strictly maintains the relative order of internal processes of the workpiece, and largely inherits the excellent genes of the parent generation, resulting in higher quality offspring;

(3) The adaptive mutation probability is used for mutation probability, which helps the algorithm to conduct fine search near the optimal solution and improve the accuracy of the solution.

However, despite the improvement of genetic algorithms, there are still some shortcomings. When generating the initial plan, only considering speed can easily result in uneven idle and busy machines. It is recommended to mix multiple rules to make the initial plan more reasonable. Lack of local search results in slow optimization in later stages. Local search operators can be added to improve search efficiency in later stages. And there are many influencing factors in actual processing, so further research and exploration are needed.

# References

[1] Mrabti, A., Bouajaja, S., & Nouri, K. (2025). Digital value stream mapping through simulation in the automotive wiring industry. *WPOM - Working Papers on Operations Management*, *16*. https://doi.org/10.4995/wpom.23686

[2] Xiao, Y., Yin, S., Ren, G., & Liu, W. (2024). Study on flexible job shop scheduling problem considering energy saving. *Journal of Intelligent & Fuzzy Systems*, *46*(3), 5493–5520. https://doi.org/10.3233/jifs-233337

[3] Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics*, *1*(1), 61–68. https://doi.org/10.1002/nav.3800010110

[4] Arthanari, T. S., & Ramamurthy, K. G. (1971). An extension of two machines sequencing problem. *Opsearch*, *8*(1), 10–22.

[5] Chryssolouris, G., Chan, S., & Cobb, W. (1984). Decision making on the factory floor: An integrated approach to process planning and scheduling. *Robotics and Computer-Integrated Manufacturing*, *1*(3–4), 315–319.

[6] Ding, B., Rao, Z., Yin, W., et al. (2024). The optimization of urban traffic routes using an enhanced genetic algorithm: A case study of Beijing South Railway Station. *Applied Sciences*, *14*(14), 6130. https://doi.org/10.3390/app14146130

[7] Naskar, A., Ghosh, S., Kundu, M., & Sarkar, R. (2025). Feature selection using guided population based genetic algorithm with modified crossover and parent selection. *Applied Soft Computing*, *172*, 112872. https://doi.org/10.1016/j.asoc.2025.112872