Journal of Engineering, Project, and Production Management
2026, 16(7), 2025-340

Green Flexible Job Shop Scheduling Using Genetic Algorithms with Adaptive Neighborhood Search

Mingyu Li[1], Lina Wang[2], Jun Wang[3], Shunwei Ma[4], and Zhiwan Liu[5]

[1] Associate Professor, School of Intelligent Manufacturing, Tianjin College, University of Science and Technology Beijing, Tianjin 301800, China

[2] Lecturer, School of Intelligent Manufacturing, Tianjin College, University of Science and Technology Beijing, Tianjin 301800, China, Email: Linawang.w@outlook.com (corresponding author).

[3] Bachelor of Engineering, Tianjin College, University of Science and Technology Beijing, Tianjin 301800, China

[4] Bachelor of Materials Science and Engineering, Tianjin College, University of Science and Technology Beijing, Tianjin 301800, China

[5] Bachelor of Engineering, Computer Science and Technology, Tianjin College, University of Science and Technology Beijing, Tianjin 301800, China

Production Management
Received December 17, 2025; revised May 26, 2026; accepted July 12, 2026
Available online July 25, 2026

Abstract: With the rise of the concept of green manufacturing, incorporating energy consumption-related objectives into scheduling problems has become an important research field. Combined with actual production scenarios, this study constructs a mathematical model for the Multi-Objective Flexible Job Shop Green Scheduling Problem (MO-FJGSP), which aims to minimize the makespan, total energy consumption, and total carbon emissions. To address the limitation of the traditional Genetic Algorithm (GA) in terms of insufficient local search capability, an Adaptive Genetic Algorithm (AGA) is designed to solve the model. A population initialization method that integrates global and local load minimization is proposed to accelerate the elimination of inferior individuals; the elite retention and roulette wheel selection strategies are combined to prevent the algorithm from falling into local optima. Simulation tests based on standard benchmark instances show that the improved GA can effectively solve the MO-FJGSP, significantly improving both the solution speed and quality. This study provides a novel methodological approach to optimizing production scheduling in green manufacturing environments.

Keywords: Multi-objective; flexible job shop; green scheduling; genetic algorithm

DOI 10.32738/JEPPM-2025-340

1. Introduction

Against the backdrop of intelligent manufacturing and Industry 4.0, the pursuit of intelligent lean production has become essential for manufacturing enterprises to strengthen their core competitiveness. Workshop scheduling, as a key method for production optimization and control, plays a crucial role in management decision-making and resource allocation. With the development of intelligent manufacturing systems, the traditional Job Shop Scheduling Problem (JSP) can no longer meet the scheduling requirements in actual production activities. Practical production scheduling needs to comprehensively consider the interactions and constraints among multiple performance indicators such as makespan, total tardiness, processing load, and cost minimization. Therefore, the complexity of solving such problems has significantly increased (Wang et al., 2021).

The MO-FJGSP is a typical problem in production scheduling. It overcomes the limitations of the specified processing machines and times in JSP and offers features such as multiple indicators, flexibility in process routes, and uncertainty in processing machines, providing greater flexibility. Each operation can be performed on different machines, and the sequence of operations for each job may also vary. Compared with the JSP, this approach aligns more closely with enterprises' actual production conditions. It is increasingly suitable for today's production mode, which involves small batches, multiple varieties, and multi-resource allocation, helping to achieve specific production goals, such as minimizing production costs and processing time. An efficient and well-designed green workshop scheduling scheme can help enterprises effectively reduce costs, conserve energy, and reduce emissions.

The MO-FJGSP is NP-hard, and traditional numerical solution methods struggle to solve it within a reasonable time (Nouiri et al., 2018). With the continuous development of artificial intelligence technology, evolutionary algorithms have become extremely effective methods for solving the MO-FJGSP (Azzouz et al., 2017), such as the weed optimization algorithm, TS algorithm, GA, DE algorithm, firefly algorithm, PSO algorithm, SA algorithm, polynomial graph algorithm, greedy algorithm, and ant colony algorithm. Intelligent algorithms can effectively schedule diverse resources and rationally allocate production processes, ultimately optimizing production performance, enhancing operational efficiency, reducing manufacturing cycles, lowering production costs, and ensuring stable production operations.

Currently, researchers have achieved research results by using these heuristic optimization algorithms. Paredis (1992) was the first to apply the GA to solve the FJSP. Luo and Qian (2020) improved the swap crossover operator and insertion mutation operator in the GA, which are prone to generating invalid solutions. They used cosine similarity to measure individual similarity, avoiding the loss of population diversity during the operation. Additionally, they applied the min-max method to constrain the multi-objective scheduling model and conducted simulations on MO-FJSP instances. Luo et al. (2019) addressed the low-carbon scheduling of the multi-objective FJSP using an improved grey wolf optimization algorithm. Huang et al. (2017) established a model for the permutation FJSP aiming to minimize makespan and total delay time, and proposed an improved chaotic weed algorithm to solve it. Amiri et al. (2019) considered uncertainty in resource allocation and job sequencing, formulated an MO-FJSP model, and solved it using a combinatorial optimization strategy. Fattahi and Fallhl (2010) considered the dynamic insertion scheduling of machines and jobs, established a dynamic scheduling model for multi-objective optimization, and used a GA to solve it. Ai and Lei (2017) used total carbon emissions as the optimization objective, formulated the corresponding low-carbon FJSP, and proposed a new shuffled frog-leaping algorithm to solve it.

GA has the advantages of simple implementation, strong versatility, fast convergence, and good robustness. Moreover, its inherent parallel characteristics enable more efficient distributed computing, making it one of the most popular algorithms for solving the MO-FJGSP at present (Brucker and Schlie, 1990). However, the traditional GA has many drawbacks, such as slow execution and a tendency to fall into local optima during later stages of the algorithm. Therefore, when applying the GA to solve the MO-FJGSP, researchers worldwide have proposed numerous improvements to address these limitations. For example, in response to the algorithm's deficiencies, new strategies have been introduced. Wang et al. (2022) proposed an adaptive NSGA-II (Non-dominated Sorting Genetic Algorithm II) in which the crossover and mutation probabilities are dynamically adjusted across different stages of the algorithm. Subsequently, they adopted the Inter-Population Crossover (IPOX) and the improved Machine Assignment and Job Processing Sequence Crossover (MPX). These two operators perform crossover operations at the operation and machine layers, respectively, thereby significantly enhancing the computational efficiency and solution accuracy of the Non-dominated Sorting Genetic Algorithm-II (NSGA-II). This approach guaranteed the optimal scheduling scheme for multi-objective flexible job shops. To address the problem of encoding and decoding affecting uniformity in subsequent genetic operations, Park et al. (2021) proposed a unified GA that enhanced the algorithm's solution efficiency. Beldjilali et al. (2014) improved the coding method by adopting an agent network structure and independently designing crossover and competition operators, thereby enhancing GA performance. Luo et al. (2020) improved the initialization method, accelerated convergence, proposed a new mutation operation, and enhanced the algorithm's performance. Zhang et al. (2021) adopted a new encoding and decoding approach, enhanced the initialization method, and employed techniques such as quantum mutation, thereby further improving algorithm performance.

In terms of integration with other algorithms, Li et al. (2020) integrated AGA with variable neighborhood search and simulated annealing and demonstrated the effectiveness of the proposed method through simulation experiments. Fekih et al. (2020) combined GA with TS to optimize the algorithm's solution performance. Burke et al. (2013) developed a genetic programming hyper-heuristic approach based on ant colony optimization for solving the hybrid flow shop scheduling problem. The method evolved scheduling rules via offline genetic programming and selects them through ant colony optimization. Su and Zhang (2017) introduced a hyper-heuristic cross-entropy algorithm to address the problem of fuzzy distributed green flow shop scheduling.

Focusing on the actual JSP, this study constructs a mathematical model for the MO-FJGSP, with optimization objectives that include minimizing makespan, total machine energy consumption and total carbon emissions. Among these objectives, energy consumption, as a green production indicator, is incorporated into the multi-objective optimization system alongside the aforementioned goals. This setting overcomes the limitation that traditional scheduling models primarily focus on time and efficiency indicators and aligns more closely with the current practical needs of green and sustainable development in the manufacturing industry. At the level of algorithm improvement, this study proposes an improved AGA to address this problem, enhancing the algorithm's ability to explore complex solution spaces and improve optimization accuracy, thereby providing a more efficient solution path for the MO-FJGSP. While ensuring makespan, this method has important theoretical significance and practical value for improving enterprise production efficiency, reducing production costs, and enhancing resource utilization.

2. Problem Description and Mathematical Model

2.1. Problem Description

MO-FJGSP is an important research topic in combinatorial optimization and production control. This problem can be described as follows: there exist *n* types of workpiece combinations, denoted as $J=\left\{J_1, J_2, \cdots, J_j, \cdots, J_n\right\}$. Each workpiece J_j needs to go through n_j processing operations, and the machine used for each operation is selected from the machine set $\left\{M_1, M_2, \cdots, M_k, \cdots, M_m\right\}$ consisting of *m* machines. The core objective of MO-FJGSP is to determine the appropriate processing machine and sequence for each operation through effective scheduling strategies to optimize makespan, total energy consumption, and total carbon emissions. Due to the greater uncertainty of the MO-FJGSP, some reasonable assumptions must be made based on the actual production process before modeling the problem.

Assumption 1: In the initial stage, the machines are idle, and all job materials have been prepared.

Assumption 2: There are precedence relationships among the operations of the same job, and these relationships will not change during production and processing.

Assumption 3: Different jobs are considered independent and equal in priority, with no constraint relationships among them. The scheduling follows the principle of minimizing overall completion time.

Assumption 4: The job's processing is continuous and uninterrupted.

Assumption 5: Each machine can process only one operation of one job at a time. There is no one-to-many relationship, and processing times may vary across machines.

Assumption 6: Machines can be in an idle state, but they cannot perform overlapping processing.

Assumption 7: Machines will not malfunction during operation.

Assumption 8: Each operation can be processed only once by a single machine, but it is eligible for assignment to one of multiple alternative machines.

2.2. Mathematical Model

Based on the research hypothesis, the symbols and definitions used to describe the MO-FJGSP are shown in Table 1. Considering the production efficiency, energy consumption, and carbon emissions factors in the manufacturing process of green flexible job shop scheduling, the optimization objectives are set to the makespan, total machine energy consumption, and total carbon emissions. Therefore, the mathematical expression of the MO-FJGSP model is constructed as follows:

Objective 1. Minimize the makespan

The makespan refers to the completion time of the last operation. It is the most fundamental optimization indicator in scheduling research and is crucial for enterprises to effectively shorten the production cycle. The completion time is calculated as shown in Eq. (2).

$$F_1 = \min\left(C_{max}\right) = \min\left(max\left(E_{jik}\right)\right) \tag{1}$$

Objective 2. Minimize the total energy consumption

The electrical energy consumed by the processing machine is divided into energy consumed during processing and during standby. This study first calculates the processing time of the workpieces and then separately calculates the electrical energy consumption during processing and standby. The sum of these two parts of electrical energy consumption is the total electrical energy consumption.

The processing time for each operation is determined by the selected processing machine, as shown in Eq. (2). The standby time before processing is given by Eq. (3).

$$X_{jik}\begin{cases}1 & T_{jik}^{p} exists\\ 0 & T_{jik}^{p} doesnotexist\end{cases} \tag{2}$$

$$T_{jik}^{s} = S_{jik} - E_{j'i'k'} \tag{3}$$

The electrical energy consumption of the processing machine in the processing and standby states is shown in Eqs. (4) and (5), respectively.

$$E_m^p = \sum_k^m \sum_j^n \sum_i^{n_j} \left(P_k^p T_{jik}^p\right) \tag{4}$$

$$E_m^s = \sum_k^m \sum_j^n \sum_i^{n_j} \left(P_k^s T_{jik}^s\right) \tag{5}$$

Then the calculation formula for total energy consumption is shown in Eq. (6).

$$F_2 = \min E_m = \min\left(E_m^p + E_m^s\right) \tag{6}$$

Table 1. Related mathematical notations and their definitions

Symbol	Definitions
n	Number of workpieces
J	Job set $\{J_1, J_2, \cdots, J_j, \cdots, J_n\}$, where J_n is the last job.
O	Operation set $\{O_{j1}, O_{j2}, \cdots, O_{ji}, \cdots, O_{jn_j}\}$, where O_{jn_j} is the last operation of the job.
O_{jik}	The i-th operation of job j is processed on machine k.
$O_{j'i'k'}$	The immediate predecessor operation of O_{jik} on machine k
m	The number of machines
M	Machine set $\{M_1, M, \cdots, M_m, \cdots, M_m\}$
M_{ji}	The optional processing machine set for operation O_{ji}
T_{jik}^{p}	The processing time of operation O_{jik}
P_k^p	The rated power of machine k in processing state
P_k^s	The rated power of machine k in standby state
F_e	Electricity emission factor
F_f	Fossil fuel emission factor
M_k^w	The mass of waste generated by machine k per unit time
F_k^m	The mass of fossil fuel required by machine k to process a unit mass of waste
ε	A very large positive number
S_{jik}	The processing start time of O_{jik}
E_{jik}	The processing completion time of O_{jik}
T_{jik}^{s}	The standby time of O_{jik} before processing; if there is no such standby time, it is 0.
X_{jik}	Binary decision variable: if the i-th operation of job j is processed on machine k, $X_{jik} = 1$; otherwise, it is 0.
X_{jik}^{w}	Binary decision variable: if the i-th operation of job j generates waste on machine k, $X_{jik}^{w} = 1$, otherwise, it is 0.
$Y_{j'i'jik}$	Binary decision variable: if Machine k processes O_{ji} immediately after finishing $O_{j'i'}$, $Y_{j'i'jik} = 1$; otherwise, it is 0.
C_{max}	Makespan
E_m	Total energy consumption
E_m^p	Energy consumption of a machine in processing state
E_m^s	Energy consumption of a machine in standby state
CE	Total carbon emissions
CE_m	Carbon emissions generated by processing machines
CE_w	Carbon emissions generated by waste
F_1	Objective function of the maximum makespan
F_2	Objective function of total machine energy consumption
F_3	Objective function of total machine carbon emissions

Objective 3. Minimize the total carbon emissions

The carbon emissions of the workshop in this study come from processing machines and the waste generated by the processing of handling machines. Among them, the electrical energy consumption of processing machines is converted into carbon emissions by multiplying it by the electrical energy emission factor, as shown in Eq. (7).

$$CE_m = E_m F_e \tag{7}$$

However, for the waste generated by handling machines, its corresponding carbon emissions need to be calculated based on the processing time during which the machine produces processing waste, combined with the waste mass generated per unit time, the mass of fossil fuels required to treat per unit mass of waste, and the fossil fuel emission factor, as shown in Eq. (8).

$$CE_w = \sum_{k}^{m}\sum_{j}^{n}\sum_{i}^{n_j}\left(T_{jik}^{p} \cdot X_{jik}^{W} \cdot M_{k}^{W} \cdot F_{k}^{m} \cdot F_f\right) \tag{8}$$

The calculation formula for total carbon emissions is given in Eq. (9).

$$F_3 = \min CE = \min\left(CE_m + CE_w\right) \tag{9}$$

Subject to (s.t.):

$$E_{jik} > S_{jik}, j \in J, i \in O, k \in M \tag{10}$$

$$E_{jik} = S_{jik} + T_{jik}^{p}, j \in J, i \in O, k \in M \tag{11}$$

$$S_{jik} = max\left\{E_{j(i-1)k}, E_{j'i'k'}\right\}, j \in J, i \in O, k \in M \tag{12}$$

$$S_{jik} > E_{j'i'k'} - \left(1 - Y_{j'i'jik}\right)\varepsilon, j, j' \in J, i, i' \in O, k \in M \tag{13}$$

$$\sum_{k=1}^{m} X_{jik} = 1 \tag{14}$$

$$\left\{X_{jik} = 1\right\} \in M_{ji}, j \in J, i \in O, k \in M \tag{15}$$

$$E_{jik} \geq S_{jik} + T_{jik}^{p}, j \in J, i \in O, k \in M \tag{16}$$

$$E_{jn_jk} \leq C_{max}, j \in J, i \in O, k \in M \tag{17}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{n_j}\left(X_{jik} \cdot T_{jik}^{p}\right) < maxE_{jik}, j \in J, i \in O, k \in M \tag{18}$$

Among them, Eq. (10) represents the sequence constraint of internal operations of a workpiece. Eq. (11) represents that the sum of the completion times of an operation is equal to the sum of the start time and processing time of the operation. Eq. (12) represents that the start time of an operation is determined by the greater value between the end time of the previous operation of the same workpiece and the end time of the immediate preceding operation of the machine processing this operation. Eq. (13) indicates that a machine can process only one operation at a time. Eq. (14) states that each operation can be processed on only one machine. Eq. (15) represents the set of processable operations for each operation. Eq. (16) represents the completion time constraint of operations. Eq. (17) represents that the completion time of each workpiece cannot exceed the total makespan. Eq. (18) represents that the accumulated working time of each machine cannot exceed the maximum completion time of that machine.

3. Adaptive Genetic Algorithm

3.1. Chromosome Coding and Decoding

The chromosomes of the GA adopt a two-segment encoding mode based on jobs and machines (Liu and Yu 2016). This encoding method can simplify the decoding process. An operation-based encoding scheme is used to determine the processing sequence of jobs. The length of the encoding equals the total number of jobs in the current batch of workpieces. Each gene in the encoding represents a workpiece number, and the number of times it appears corresponds to the operation number for that job. Machine-based encoding is used to obtain the processing machine for each operation. The encoding sequence is arranged according to the processing order of the jobs, but does not directly correspond to the operation encoding.

Table 2 illustrates the flexible scheduling of 3 workpieces on 4 machines. Fig. 1 shows an encoding example for the MO-FJGSP solution corresponding to Table 2, which represents a feasible solution to the optimization problem. The chromosome length is equal to twice the total number of operations. In this instance, there are 9 operations in total, so the chromosome length is 18 genes. When scanning the operation sequence, in the workpiece layer from left to right, the first “1” refers to the first operation of the workpiece to be processed on the machine, corresponding to “2” in the machine allocation part; the second “1” represents the second operation of workpiece 1 to be processed on the machine, corresponding to “1” in the machine allocation part; the third “1” indicates the third operation of workpiece 1 to be processed on the machine, corresponding to “2” in the machine allocation part. According to the above description, the operation sequence for the machines M_1 is $O_{21} \rightarrow O_{12}$, the machines M_2 are $O_{11} \rightarrow O_{13} \rightarrow O_{33}$, the machines M_3 are $O_{32} \rightarrow O_{22} \rightarrow O_{34}$, and the machine M_4 is O_{31}. The solved operation sequence and machine allocation can be interpreted as follows: {(O_{11}, M_2), (O_{12}, M_1), (O_{31}, M_4), (O_{21}, M_1), (O_{13}, M_2), (O_{33}, M_2), (O_{32}, M_3), (O_{22}, M_3), (O_{34}, M_3)}.

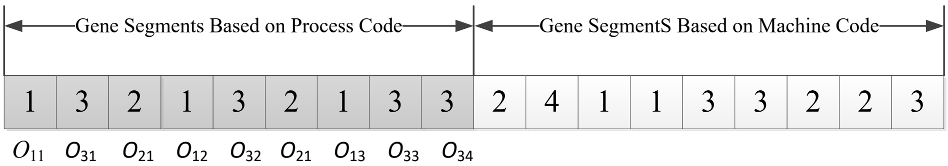


Fig. 1. Chromosome coding example

Table 2. Flexible scheduling of three workpieces on four machines

Workpieces	Process	Processing Time/min			
		M1	M2	M3	M4
J1	O11	14	10	11	--
	O12	--	--	9	5
	O13	9	12	12	18
J2	O21	11	8	--	--
	O22	18	10	20	14
J3	O31	15	--	8	18
	O32	--	12	--	10
	O33	14	10	13	--
	O34	6	--	4	--

Decoding refers to assigning operations to the corresponding machines based on the chromosome encoding in order to obtain a feasible scheduling solution. The decoding process is illustrated in Fig. 2, from which the processing machine and processing time for each operation can be derived. The optimized scheduling Gantt chart, generated from the chromosome encoding shown in Fig. 1, is presented in Fig. 3.

3.2. Population Initialization

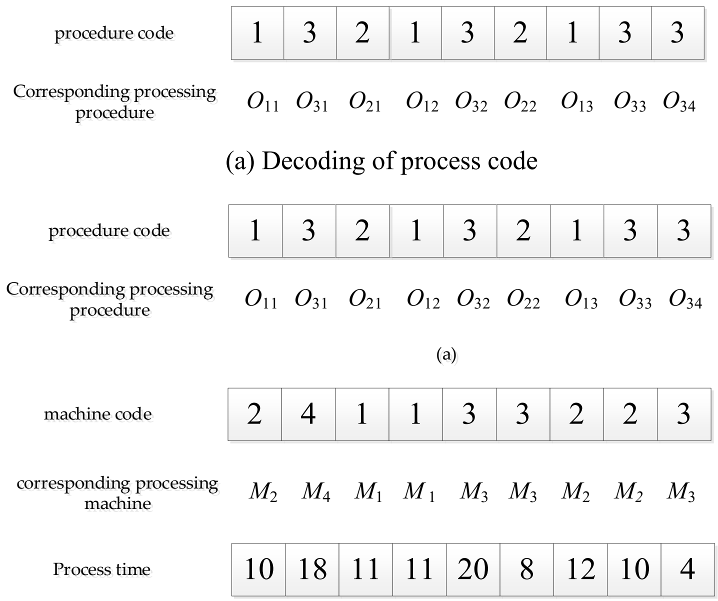
The quality of the initial population has a substantial influence on the solving speed and quality of the GA. Currently, the random population initialization method is commonly employed. However, it often results in a low-quality initial solution, an extended solving time, or even the failure to obtain the optimal solution, resulting in imbalanced machine loads. This study puts forward an initialization method based on the minimum overall load and the minimum local load. For the machine encoding, it is initialized according to two selection strategies: the shortest overall time and the shortest local workpiece time. The encoding of the operation part in the initial population is generated randomly.

The minimum overall load strategy implies that when each operation chooses a machine, it takes into account the available processing time of the current machine. Under the constraint of preserving the operation sequence, the machine with the shortest overall processing time is selected. The minimum local load strategy refers to selecting, for each operation, the machine that requires the least processing time. This can effectively enhance the quality of the initial solution for the machine-selection part, ensure the randomness and diversity of the initial population, and accelerate the convergence rate of the algorithm, as demonstrated in Tables 3 and 4.

3.3. Evaluation of Fitness Function

The fitness function of GA can quantify an individual's adaptability, evaluate its quality, retain and develop the more fit individuals in the chromosome, and serve as the basis for selection and evolution. It serves as an important component of the genetic algorithm. Eq. (19) represents the normalization processing of the objective function.

$$F_{ii}^{*}=\left|\frac{F_{ii}-F_{\min ii}}{F_{\max ii}-F_{\min ii}}\right|,\quad ii=1,2,3 \tag{19}$$



(b) Decoding of machine code.

Fig. 2. Chromosome decoding example

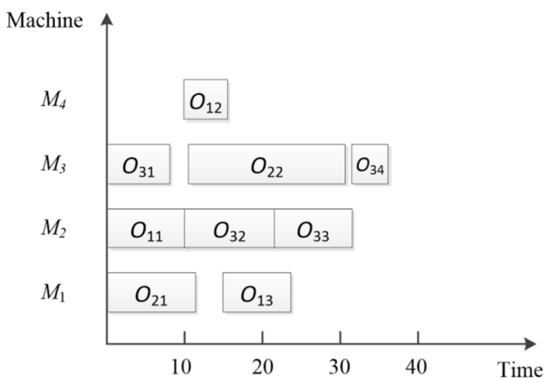


Fig. 3. Gantt chart of the decoded scheduling solution

In most cases, the GA calculates the fitness value as the reciprocal of the chromosome's objective function value. The higher the fitness value, the better the individual. The fitness function is expressed in Eq. (20).

$$f\left(i\right)=\frac{1}{\alpha F_{1}^{*}+\beta F_{2}^{*}+\gamma F_{3}^{*}} \tag{20}$$

Where: F_{ii}^{*} Normalized objective function value; $F_{\max ii}$ represents the global target maximum; $F_{\min ii}$ represents the minimum value; F_{ii} represents the current target value; $f\left(i\right)$ represents the fitness value; α,β,γ represents the weight coefficient, and $\alpha+\beta+\gamma=1$

Assuming that decision-makers have high requirements for environmental indicators and hope to obtain environmentally friendly decisions (Zhang and Shao, 2023). The selected weight coefficient matrix is [α, β, γ] = [0.25, 0.25, 0.5].

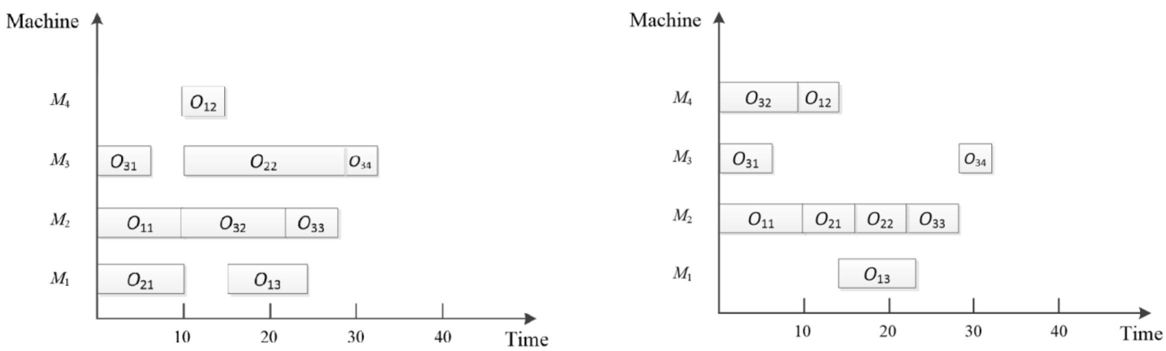
When the fitness values of the chromosomes are the same, chromosomes with a lower machine load balance degree are more likely to be inherited by the next generation, because they have more scheduling space for operations to optimize fitness. Fig. 4 shows Chromosomes A and B with the same fitness value. Chromosome B has a lower degree of machine load balancing, resulting in more idle time on the machine tools for optimization. Therefore, Chromosome B has a higher quality value.

Table 3. Overall load minimum selection strategy

Machine processing time	[0 0 0 0]	[2 0 0 0]	[2 4 0 0]	[2 4 0 4]
process	O11→	O12→	O21→	O22
Optional machine set	M1 M2 M3 M4	M2 M3	M1 M2 M3 M4	M1 M2 M4
Processing time	[2 6 3 4]	[4 6]	[5 3 7 4]	[6 5 8]
Machine selection	M1	M2	M4	M1
processing time array	[2 0 0 0]	[2 4 0 0]	[2 4 0 4]	[8 4 0 4]
Machine coding	[1 0 0 0]	[1 2 0 0]	[1 2 4 0]	[1 2 4 1]

Table 4. Local load minimum selection strategy

Procedure code	[1 0 0 0]	[1 2 0 0]	[1 2 2 0]	[1 2 2 1]
process	O11→	O21→	O22→	O12
Optional machine set	M1 M2 M3 M4	M1 M2 M3 M4	M1 M2 M4	M2 M3
Processing time	[2 6 3 4]	[5 3 7 4]	[6 5 8]	[4 6]
Machine selection	M1	M2	M2	M1
processing time array	[2 0 0 0]	[2 4 0 0]	[2 4 0 4]	[8 4 0 4]
Machine coding	[1 2 2 1]			



(a) Chromosome A (b) Chromosome B

Fig. 4. Load balancing degree of the same fitness chromosome

3.4. Selection Operator

To enable the GA to develop in a more advantageous direction in the early stages of iteration, this paper proposes an improved selection strategy. The selection operation determines the selection probability based on individuals' fitness values. If the fitness value of a new individual exceeds that of its parent, the individual is retained. Then, together with the remaining individuals, a combined elitism and roulette wheel selection method is applied. The two best and second-best individuals among the retained new individuals and the parent individuals are directly copied to the offspring, while the remaining individuals are selected via roulette-wheel selection. This ensures the quality of the selected individuals, prevents the loss of genes of excellent individuals due to random selection, and tries to retain more excellent individuals. The specific steps in the operation are shown below.

Step 1: Calculate the fitness value $f(i)$ (i =1, 2, …, N) according to Eq. (20), where N represents the number of chromosomes in the population.

Step 2: Taking the fitness value of the chromosome as the basis for calculation, calculating the genetic probability $P(i)$ according to Eq. (21).

$$P(i) = \frac{f(i)}{\sum_{j=1}^{N} f(j)} \tag{21}$$

Where: $f(i)$ represents the fitness value of each individual in population f_i; $\sum_{j=1}^{N} f(j)$ represents the total fitness of population f.

Step 3: Based on the genetic probability $P(i)$ of the chromosomes, the cumulative probability $q(i)$ can be obtained. The expression is shown in Eq. (22).

$$q(i)=\sum_{j=1}^{i}P(j) \tag{22}$$

Step 4: If $rand < q(1)$, then select individual 1; otherwise, select individual k such that the equation $q(k-1) < rand < q(k)$ holds. Repeat this operation *N* times in total.

3.5. Cross Operator

The traditional GA typically uses a fixed crossover probability. In the early stages of evolution, the population contains a wide variety of individuals but relatively few high-quality solutions. A relatively large crossover probability can enhance the algorithm's local search capability and increase population diversity. However, in the later stage of evolution, when an individual's fitness exceeds the population's average fitness, maintaining a high crossover probability may lead to blind search behavior, slow convergence, or even algorithm stagnation. At this point, a lower crossover probability should be adopted to accelerate convergence and preserve high-quality solutions. To address this issue, the Adaptive Genetic Algorithm (AGA) was introduced.

This algorithm divides the population into a high-quality set and a low-quality set according to the fitness values $f(i)$ of individuals. Let f_a be the average fitness value of the population, which k_{mc} is represented by a quality coefficient. For chromosomes in the high-quality set, $k_{mc} > 0$, and for chromosomes in the low-quality set, $k_{mc} < 0$. The crossover probability between high-quality chromosomes decreases, while the crossover probability between low-quality chromosomes increases, thereby accelerating the generation of high-quality solutions (Zhang and Mei, 2019). The improved crossover probability is shown in Eq. (23).

$$p_{mc} = p_{mc0} - \frac{\left(f_a - f_a^{'}\right)k_c}{f_a^{'}} - \left(k_{mc1} + k_{mc2}\right) \tag{23}$$

Where: p_{mc0} is the initial crossover probability; $f_a^{'}$ is the average fitness value of the initial population; k_c is the crossover coefficient; k_{mc1}、k_{mc2} are the quality coefficients of the chromosomes undergoing crossover, respectively.

3.6. Mutation Operator

The mutation operation helps maintain population diversity to a certain extent. Traditional mutation operations are applied with a fixed mutation probability, which may disrupt effective genes in high-quality individuals, leading to premature convergence and reducing the algorithm's ability to find the optimal solution in later stages. To improve the guidance and accuracy of the mutation operation, a lower mutation probability should be applied to high-quality individuals to preserve their excellent genes and further protect them in the population. For low-quality individuals, a higher mutation probability should be used to make them more adaptable to survival. Therefore, a non-uniform mutation rate P_m is introduced, which increases the GA's mutation rate in the initial stage to more comprehensively and quickly explore the solution space. As the number of iterations increases, the mutation rate is gradually reduced to enable more precise exploration of the optimal solution and to achieve better optimization results**.** The expression for the non-uniform mutation rate is given by Eq. (24).

$$P_m = P_{m_0} + \rho\left(1 - \frac{gen}{\max gen}\right) \tag{24}$$

Where: P_{m_0} is the initial mutation rate. The range of ρ is (0,1). The function of ρ is to control the rate of decrease in the mutation rate. When the value of ρ is close to 1, the decreasing speed of the mutation rate will be relatively slow; when the value of ρ is close to 0, the rate of decrease in the mutation rate will be relatively fast. For the mutation of workpiece encoding, a neighborhood search-based mutation operator is employed, while a single-point mutation method is used for machine encoding. The corresponding operational diagrams are presented in Fig. 5 and Fig. 6.

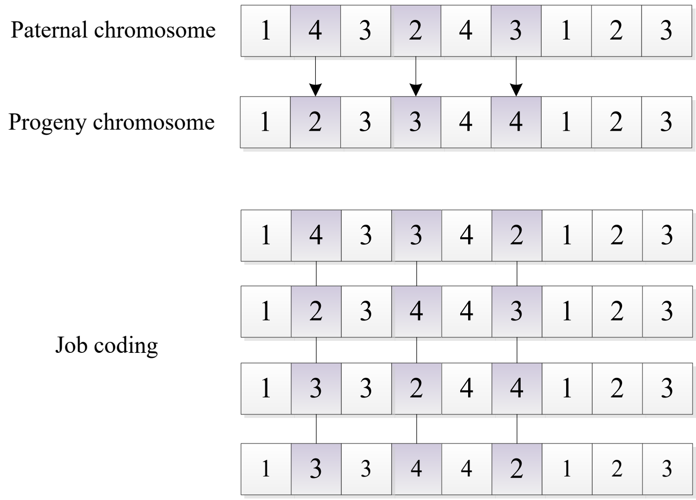


Fig. 5. Variation operations based on neighborhood operations at the process level

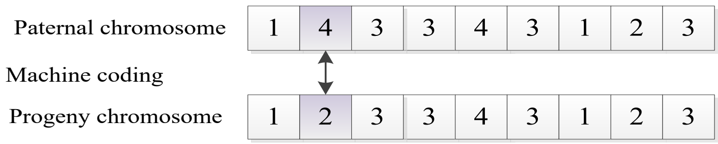


Fig. 6. Machine coded single point mutation operation

3.7. AGA Process

To address the problem that the traditional GA tends to lose high-quality solutions and easily fall into local optima, this study adopts the roulette-wheel selection method to accelerate the elimination of inferior genes, thereby improving the population's convergence speed.

Step 1: Set the relevant parameters and randomly generate an initial population.

Step 2: Encoding and decoding.

Step 3: Adopt the elitist retention roulette wheel selection strategy to select excellent individuals.

Step 4: Calculate the fitness values of the individuals in the population.

Step 5: Perform a uniform crossover on the selected parent generations to update individuals' encodings, then screen them using the non-uniform crossover rate.

Step 6: Conduct a neighborhood search mutation operation on the workpiece encoding, and adopt a single-point mutation for the machine encoding. Then apply a non-uniform mutation rate to improve the algorithm's search capability and stability.

Step 7: Determine whether the termination condition is met. If the maximum number of iterations is reached or the stop condition is met, terminate the iterative process; otherwise, return to Step 4.

Step 8: Output the results.

4. Computational Experiments

In this study, MATLAB 2021 is used for simulation, and the simulation is run on a PC with Windows 10.0, an Intel Pentium CPU 4415U processor at 2.30 GHz, and 4 GB of RAM. The basic parameter settings of the improved AGA are shown in Table 5.

4.1. Performance Analysis of Models

To further verify the performance of the Improved AGA algorithm proposed in this study, tests were conducted on three benchmark cases (8×8, 10×10, and 15×10) proposed by Kacem, and comparisons were made with the multi-objective branch and bound (MBB) algorithm proposed by Soto C (2020) for solving multi-objective integer programming problems, as well as the GA. The optimization objectives are to minimize the maximum completion time (F_1), the total energy consumption of machines (F2), and the total carbon emissions (F3). The comparison of the case results is shown in Table 6.

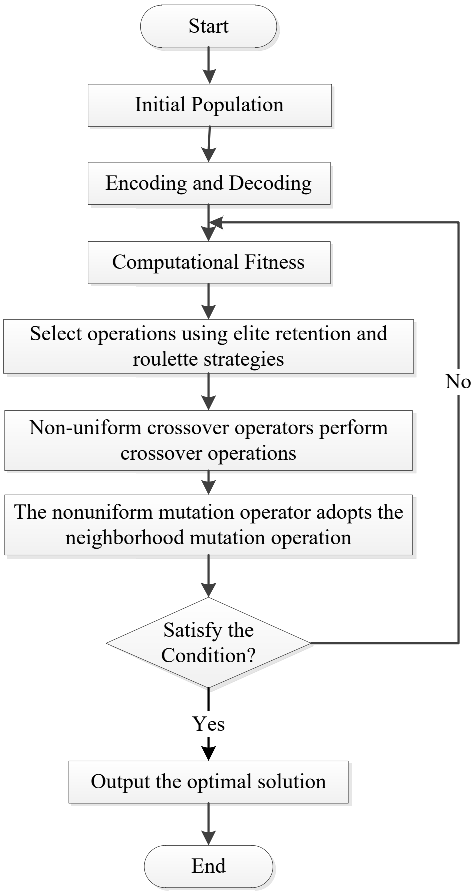


Fig. 7. Improved flow chart of GA

Table 5. Configuration of example parameters

Parameter	Value	Parameter	Value
Population size (N)	100	$P_{c\max}$	0.9
Number of iterations max *gen*	250	$P_{c\,\mathrm{mid}}$	0.6
$P_{m\max}$	0.2	$P_{c\min}$	0.4
$P_{m\,\mathrm{mid}}$	0.05	$P_{m\min}$	0.001
P_c	0.8	P_m	0.2
Load balancing factor	5	Initial crossover probability $P_{\mathrm{mc}0}$	0.8
k_1 k_c	0.42	k_{mc1}	0.1
k_{mc2}	-0.1	$P_{\mathrm{m}v0}$	0.03
k_v	0.03	k_2 Cross probability adjustment parameter	0.7
Variation probability adjustment parameter ρ	0.6		

In Table 6, the number of non-dominated solutions obtained by the AGA algorithm when solving the 8×8 case is 4, which is more than those obtained by MBB and GA. Moreover, data analysis shows that [15,13,73] and [16,12,75] outperform the GA solution sets [17,15,73] and [16,13,79]. In the 10×10 case, the number of non-dominated solutions obtained by AGA exceeds those of both MBB and GA. For the 15×10 case, the number of non-dominated solutions obtained by AGA is 1 more than that of MBB and GA. It can be concluded that the AGA algorithm can effectively solve multi-objective optimization problems of varying scales and obtain high-quality solution sets.

Table 6. Comparison table of Kacem example results

Case	Obj.	AGA				MBB			GA	
8×8	F1	15	16	16	17	15	17	14	17	16
	F2	13	11	12	11	13	11	10	15	13
	F3	73	77	75	75	73	75	78	73	79
10×10	F1	8	7	7	-	8	7	-	9	-
	F2	6	6	5	-	6	7	-	8	-
	F3	40	41	43	-	43	42	-	40	-
15×10	F1	16	12	-	-	12	-	-	15	-
	F2	10	10	-	-	11	-	-	12	-
	F3	91	93	-	-	91	-	-	93	-

4.2. Case Application Analysis

Two cases of different scales, 6×6 and 6×8, are selected from the reference (Zhu 2024). The data used is a partial dataset from the frame longitudinal beam processing operations of an automobile parts manufacturing enterprise (Zhao et al., 2025), and was solved by AGA/GA. The AGA algorithm is compared with the traditional GA to verify its effectiveness. To improve the credibility of the comparison results, the basic parameters of the two algorithms are held constant.

Example 1 is a 6×6 example composed of 6 machines and 6 workpieces to be processed, and it is a typical MO-FJGSP problem. The AGA finds that the makespan is 9, the total energy consumption is 7, and the total carbon emissions are 40. The corresponding convergence graph and Gantt chart of this solution are shown in Figs. 8 and 9. In Fig. 8, the vertical axis, ObjV（F1）, represents the completion time, while the horizontal axis (gen) denotes the number of generations.

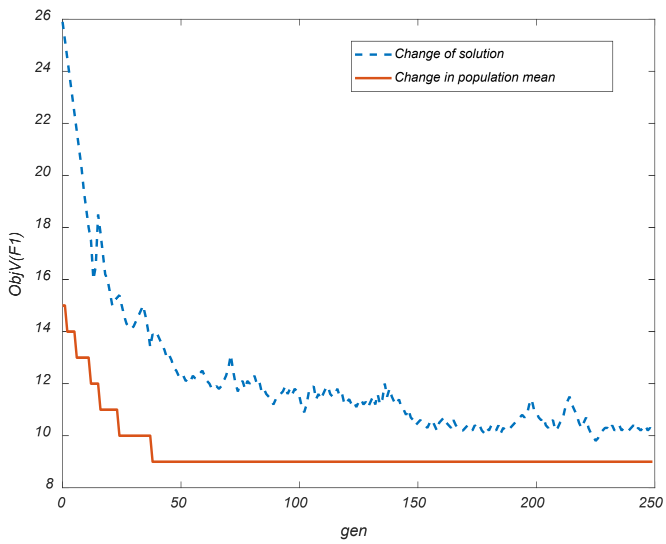


Fig. 8. Improved AGA algorithm for convergence of 6×6 results (F1)

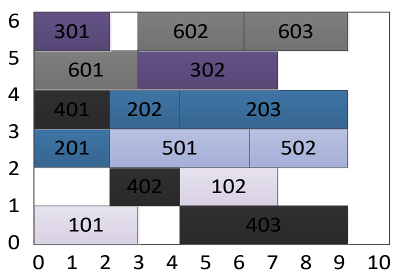


Fig. 9. Improved AGA algorithm for 6×6 Gantt chart results

Example 2 is a set of 6×8 examples. The value solved by the AGA algorithm is [53 240 256]. The corresponding convergence graph and Gantt chart of this value are shown in Figs. 10 and 11.

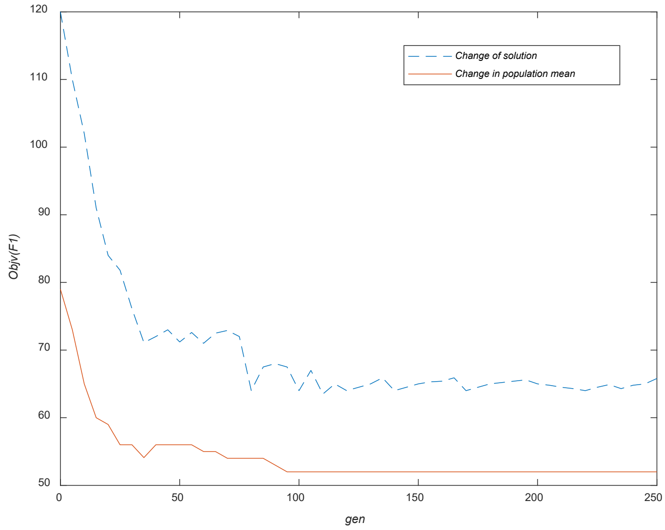


Fig. 10. AGA 6×8 result convergence graph (F1)

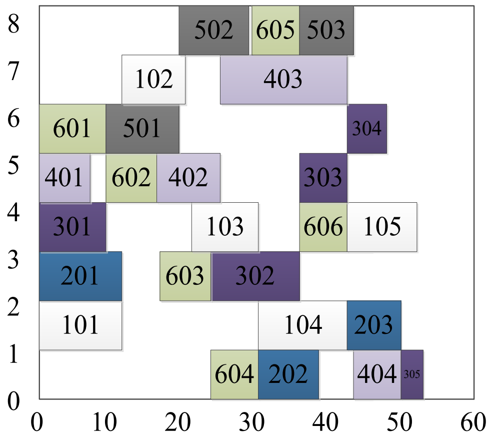


Fig. 11. The Gantt chart with 6×8 result of AGA is improved

The optimal solutions obtained by the three algorithms are presented in Table 7. While "Average" represents the mean value of the solutions obtained over ten independent runs.

Table 7. Improved algorithm example result data comparison table

algorithm	6×6 cases		6×8 cases	
	Optimal Results	Average Results	Optimal Results	Average Results
Zhu (2024)	11	10.6	55	56.9
GA	11	11.4	57	60.4
AGA	9	9.8	53	55.2

In Table 7, for the solution of the 6×6 case, the optimal solution obtained by the traditional GA is 11, while the optimal solution obtained by the improved AGA proposed in this study is 9. Compared with the traditional GA, the optimization accuracy is improved by 18.2%. For the 6×8 case, the optimal solution reported by Zhu et al. (2024) is 55, while the optimal value obtained by the AGA proposed in this study is 53. Here, the "optimal solution" denotes the best solution found by the algorithm. These experimental data demonstrate the effectiveness of the improved algorithm in this study for solving the FJSP.

The AGA converges relatively quickly in the early stages, indicating that the crossover mechanism effectively enhances the algorithm's iteration efficiency. In the later stages of the algorithm's iterations, the adaptive adjustment of crossover and mutation probabilities enables the improved algorithm to escape local optima and obtain the current optimal solution for the instance. When solving the 6×6 case, compared with the traditional GA, the AGA achieves a similar number of generations to converge. However, the improved algorithm in this study has produced better offspring solutions in the same number of genetic generations, demonstrating a significantly improved optimization efficiency.

5. Conclusion

In this study, a flexible job-shop green scheduling model is developed with multiple objectives, including minimizing the make-span, total machine energy consumption and total carbon emissions. To solve this problem, an improved AGA is proposed. This algorithm is improved in aspects such as coding methods, population initialization, crossover and mutation operations, and selection operations. It effectively improves the quality of the initial solution, prevents the algorithm from falling into a local optimum in the later stage of evolution, enhances the global search ability of the algorithm, resulting in faster convergence, and can be adapted to diverse production scenarios across different enterprises by adjusting the weight coefficients of cost and time. Managers can shift their scheduling decisions from simply pursuing completion time to a multi-objective trade-off, flexibly selecting solutions that prioritize efficiency, energy saving, or emission reduction using the Pareto solution set output by the model and adjustable weights. In machine allocation and process scheduling, the initialization and adaptive strategies proposed in this study can replace experience-based scheduling, reducing idle time and energy waste. Simultaneously, processing energy consumption, standby energy consumption, and waste carbon emissions can be incorporated into daily assessments, ensuring that scheduling schemes balance delivery cycles and environmental footprint.

To verify the effect of the AGA, classic FJSP examples are selected for comparative experiments. Through comparison of Gantt charts and convergence graphs, it is clear that the improved algorithm is superior to the traditional algorithm in both convergence speed and the best solution found, thus verifying the feasibility and effectiveness of the improvement. However, this study focuses solely on static scheduling problems; further research is needed to extend the approach to more complex scenarios, such as dynamic rescheduling.

Despite the effectiveness of the proposed Adaptive Genetic Algorithm (AGA), we acknowledge certain limitations, particularly in preserving population diversity. This limitation may affect the algorithm's ability to explore the solution space comprehensively in multi-objective scenarios. In future work, we plan to integrate Pareto-based selection mechanisms, such as non-dominated sorting (e.g., NSGA-II) and crowding-distance techniques, to enhance both the convergence behavior and the maintenance of diversity of the algorithm. We believe these improvements will further strengthen the robustness and applicability of the proposed approach in solving complex multi-objective scheduling problems.

Author Contributions

Mingyu Li contributed to conceptualization, methodology, validation, formal analysis, software, writing, review and editing. Lina Wang contributed to conceptualization, methodology, formal analysis, and writing, review and editing. Jun Wang contributed to conceptualization, formal analysis, validation, writing, review and editing. Shunwei Ma contributed to methodology. Zhiwan Liu contributed to investigation. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the Scientific Research Project of Tianjin Education Commission (Grant No.2023KJ251) “Research and Development of an Integrated Maintenance and Inspection Vehicle for Photovoltaic Panels”. Additional support was provided by the Fourth Batch of First-Class Course Construction Projects of Beijing University of Science and Technology, Tianjin College (Grant No.YLKC202413) “Principles of Automatic Control A”.

Institutional Review Board Statement

Not applicable.

Declaration of Artificial Intelligence (AI) Tools

The authors used Grammarly Premium solely for language editing and readability improvement. The authors reviewed and verified all content and take full responsibility for the accuracy and integrity of the manuscript.

References

Ai, Z.Y. and Lei, D. M. (2017). A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling. *Control Theory & Applications,* 34(10), 104-111. doi: 10.7641/CTA.2017.60768

Amiria, F., Babak, S., and Ali, T. (2019). Multi-objective simulation optimization for uncertain resource assignment and job sequence in automated flexible job shop. *Applied Soft Computing Journal,* 75, 190-202. doi : 10.1016/j.asoc.2018.11.018

Azzouz, A., Ennigrou, M., and Said, L.B. (2017). A self-adaptive hybrid algorithm for solving flexible job-shop problem with sequence dependent setup time. *ScienceDirect,* 112, 457-466. doi: 10.1016/j.procs.2017.08.023

Beldjilali, B., Aissani, N., and Meziane, M. E. A. (2014). Hybrid genetic algorithm for flexible job-shop scheduling problem. *The Mediterranean Journal of Computers & Networks,* 10(3), 267-274.

Brucker, P. and Schlie, R. (1990). Job-shop scheduling with multi-purpose machines. *Computing, 45*(4), 369-375. doi: 10.1007/BF02238804

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Qu, R. (2013). Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society,* 64(12), 1695-1724. doi: 10.1057/jors.2013.71

Fattahi, P. and Fallhl, A. (2010). Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability. *ClRP Journal of Manufacturing Science and Technology*, 2(2), 114-123. doi: 10.1016/j.cirpj.2009.10.001

Fekih, A., Hadda, H., Kacem, I., and Hadj-Alouane, A. B. (2020). A hybrid genetic Tabu search algorithm for minimising total completion time in a flexible job-shop scheduling problem. *European Journal of Industrial Engineering,* 14(6), 763-781. doi: 10.1504/EJIE.2020.112479

Huang, X., Ye, C.M., and Cao, L. (2017). Chaos invasive weed optimization algorithm for multi-objective permutation flow shop scheduling problem. *Systems Engineering-Theory & Practice,* 37(1), 253-262. doi: 10.12011/1000-6788(2017)01-0253-10

Li, S.H., Hu, R., Qian, B., Zhang, Z. Q., and Jin, H. P. (2020). Hyper-heuristic genetic algorithm for fuzzy flexible job shop scheduling. *Control Theory and Applications,* 37(2), 316-330. doi: 10.7641/CTA.2019.80813

Liu, S., and Yu, H.Q. (2016). Research on multi-objective FJSP problem based on improved genetic algorithm. *Control Engineering of China,* 23(6), 816-822. doi: 10.14107/j.cnki.kzgc.150184

Luo, S., Zhang, L. X., and Fan, Y. S. (2019). Energy-efficient scheduling for multi-objective flexible job shops with variable processing speeds by grey wolf optimization. *Journal of Cleaner Production*, 234, 1365-1384. doi: 10.1016/j.jclepro.2019.06.151

Luo, W., Qian, Q., Fu, Y. F, and Feng, Y. (2021). Solution to Multi-Objective Flexible Job Shop Scheduling Problem Based on Constraint Model. *Process Automation Instrumentation,* 42(07), 37-41+46. doi: 10.16086/j.cnki.issn1000-0380.2020100047

Luo, X., Qian, Q., and Fu, Y.F. (2020). Improved genetic algorithm for solving flexible job shop scheduling problem. Procedia Computer Science, 166, 480-485. doi: 10.1016/j.procs.2020.02.065

Nouiri, M., Bekrar, A., Jemai, A., Niar, S., and Ammari, A. C. (2018). An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *Journal of Intelligent Manufacturing*, 29(3), 603-615. doi: 10.1007/s10845-015-1039-3

Paredis, J. (1992). Exploiting Constraints as Background Knowledge for Genetic Algorithms: A Case-Study for Scheduling. ***In*** *PPSN*, 231-240,

Park, J. S., Ng, H. Y., Chua, T. J., Ng, Y. T., and Kim, J. W. (2021). Unified genetic algorithm approach for solving flexible job-shop scheduling problem. *Applied Sciences,*11 (14), 6454. doi: 10.3390/app11146454

Soto, C., Dorronsoro, B., and Fraire, H. (2020). Solving the multi-objective flexible job shop scheduling problem with a novel parallel branch and bound algorithm. *Swarm and Evolutionary Computation*, 53, 100632. doi: 10.1016/j.swevo.2019.100632

Su, N. and Zhang, M. J. (2017). A PSO-based Hyper-heuristic for Evolving Dispatching Rules in Job Shop Scheduling. *2017 IEEE Congress on Evolutionary Computation(CEC)*, Donostia-San Sebastian, Spain, 5-8 June. doi: 10.1109/CEC.2017.7969402

Wang, C. Y., Li, Y., and Li, X. Y. (2021). Solving flexible job shop scheduling problem by a multi-swarm collaborative genetic algorithm. *Journal of Systems Engineering and Electronics, 32*(2), 261-271. doi: 10.23919/JSEE.2021.000023

Wang, G., Gao, S., and Fang, S. J. (2022). Multi-Objective Flexible Job Shop Scheduling Based on Adaptive NSGA-II. *Machine Tool & Hydraulics ,* 50(18), 129-135. doi: 10.3969/j.issn.1001-3881.2022.18.025

Zhang, J. C., Shao, L. Z, and Lei, X. M. (2023). A multi-objective green flexible job shop scheduling method based on an improved NSGA-II algorithm. *Manufacturing Technology & Machine Tool,* (01), 145-152. doi: 10.19287/j.mtmt.1005-2402.2023.01.024

Zhang, S. J., Du, H. T., Borucki, S., Jin, S., Hou, T., and Li, Z. (2021). Dual resource con-strained flexible job shop scheduling based on improved quantum genetic algorithm. *Machines*, 9(6), 1-8. doi: 10.3390/machines9060108

Zhang, Z. W., Mei, H. Y., Zhou, J., Jia, H. P. (2019). A rule extraction method based on multi-objective co-evolutionary genetic algorithm. *Journal of Shandong University (Engineering Science*), 49(2), 122-130. doi: 10.6040/j.issn.1672-3961.0.2018.211

Zhao, X. Y., Miao, H. B., Zhang, S. B., Liu, H. J., and Wu, Y. D. (2025). Multi-objective green flexible job shop scheduling based on improved dual population genetic algorithm. *Industrial Engineering Journal,* (07), 1-16. doi: 10.3969/j.issn.1007-7375.250023

Zhu, M. J. (2024). CA-type neighborhood genetic algorithm for flexible job shop scheduling problem. *Information Technology and Informatization,* (07), 82-86. doi: 10.3969/j.issn.1672-9528.2024.07.017

Mingyu Li obtained his PhD in Operations Research (2010) from the Rocket Force University of Engineering in Xi'an. Presently, he is working as an Associate Professor in the Department of Tianjin College at the University of Science and Technology Beijing, Tianjin. He has published articles in more than 2 internationally reputed peer-reviewed journals and conference proceedings. His areas of interest include smart manufacturing, logistics optimization, and education and teaching.

Lina Wang obtained her Master of Engineering (2020) from Changchun University of Science and Technology. She is currently working as a Lecturer in the Intelligent Manufacturing program at Tianjin College, University of Science and Technology Beijing. She has published articles in more than 2 internationally reputed peer-reviewed journals and conference proceedings. Her research interests include intelligent manufacturing and intelligent control. She is committed to advancing academic development in related fields.

Jun Wang obtained his Bachelor of Engineering degree in Automation from Tianjin College, University of Science and Technology Beijing in 2026. As an outstanding undergraduate with excellent academic performance, he received the National Encouragement Scholarship and was awarded the honorary title of Excellent Student of Tianjin. He served as the project leader of a provincial-level college student innovation program and the person in charge of a national-level crowd innovation space laboratory. He has won more than 30 awards in various discipline competitions at national, provincial and municipal levels. His research interests include machine learning, image processing, pattern recognition and intelligent control. He has a solid theoretical foundation and rich practical experience in engineering projects, and is committed to further study and research in related fields.

Shunwei Ma obtained a bachelor's degree in Materials Science and Engineering from Beijing University of Science and Technology, Tianjin College in 2026. As an outstanding undergraduate with excellent academic performance, he received the second-class scholarship and was honored with the title of "Outstanding Class Leader". He has served as the project leader for a national undergraduate innovation project and as the laboratory director of a national innovation space. He has accumulated over twenty awards in various national, provincial and municipal academic competitions. He published "Potassium-Based Solid Sorbents for CO2 Adsorption: Key Role of Interconnected Pores" as the co-first author in the journal "Nanomaterials". He also published "Application Progress of Magnetic MOF Materials in Water Treatment" as the first author in the journal "Chemical Engineering Design Communications". His main research directions include nanomaterials and heat treatment of metal materials. He has a solid theoretical foundation in his field and rich practical experience in engineering projects. He is committed to furthering his studies and research in related fields.

Zhiwan Liu obtained a Bachelor of Engineering degree in Computer Science and Technology from Tianjin College, University of Science and Technology Beijing in 2025. He holds an Intermediate Software Designer Certification. His notable achievements include the First Prize in the 2024 2nd "Huashu Cup" International Mathematical Contest in Modeling, the Third Prize in the 2023 9th National Undergraduate Physics Experiment Competition, the National Scholarship for the 2023 - 2024 academic year, the National Encouragement Scholarship for the 2022 - 2023 academic year, the title of "Tianjin Outstanding Student" for the 2022 - 2023 academic year, the "Outstanding Merit Student" award for the 2023 - 2024 academic year, and the "Outstanding Graduate" honor in 2025. His primary research interests encompass machine learning, algorithm optimization, and applications of mathematical modeling. With a solid theoretical foundation and extensive practical experience in disciplinary competitions, he is committed to pursuing further studies and research in related fields.

Abbreviations

The following abbreviations are used in this manuscript

JSP	Job Shop Scheduling Problem
MO-FJGSP	The multi-objective flexible job shop green scheduling problem
MO-FJSP	The multi-objective flexible job shop scheduling problem
TS	Tabu search algorithm

GA	Genetic algorithm
DE	Differential evolution algorithm
PSO	Firefly algorithm, particle swarm optimization algorithm
SA	Simulated annealing algorithm
AGA	Adaptive Genetic Algorithm
FJSP	Flexible Job Shop Scheduling Problem