Multi-layer collaborative optimization method for solving fuzzy multi-objective integrated process planning and scheduling

Xiaoyu Wen, Xiaonan Lian, Kanghong Wang, Hao Li and Guofu Luo

Abstract
Research on integrated process planning and scheduling (IPPS) is of great significance to the improvement of the overall quality of machinery manufacturing system. In the actual manufacturing process, the manufacturing system is often accompanied by some unpredictable uncertain disturbance factors, for instance uncertain processing time of jobs and changes of due date, etc. These uncertain disturbance events will ultimately affect production efficiency and customer satisfaction. Consequently, this paper considers the multi-objective IPPS problem with uncertain processing time and uncertain due date. A multi-layer collaborative optimization (MLCO) method is designed for the fuzzy multi-objective IPPS (FMOIPPS) problem, including three layers. For the process planning layer, the basic genetic algorithm is used to provide various near optimal process plans for the process selection system. For the process selection layer, a multi-objective genetic algorithm (MOGA) is designed to optimize the process selection population. A sharing function method is introduced to maintain population diversity. An individual comprehensive evaluation method is introduced to evaluate non-dominated solutions. The crowded distance, fast non-dominated sorting and elite strategy based on NSGAII is adopted in the proposed MOGA. The external archive method is employed to preserve the non-dominated solutions generated during population evolution. For the scheduling layer, a MOGA with a boundary search strategy is proposed. The boundary search strategy is designed to improve the search ability of boundary solutions. Three optimization objectives are minimizing the spread of fuzzy makespan, minimizing fuzzy makespan and maximizing average customer satisfaction simultaneously. The target of scheduling layer is to make scheduling arrangements for the process information obtained by process selection layer. Through mutual cooperation among each layer, guide the overall optimization process, and finally get satisfactory solutions. Different problem examples of various scales are employed to verify feasibility and effectiveness of the MLCO method. The experimental results indicate that the MLCO method can effectively address FMOIPPS problem.

Keywords
Integrated process planning and scheduling, fuzzy multi-objective optimization, sharing function, boundary search strategy

Introduction
Process planning and scheduling are important links in manufacturing system which are inseparable for the development of advanced manufacturing industry.1,2 With the development of computer technology, process planning and scheduling began to develop from artificial experience to precise planning and scientific decision-making of computer-aided. In the actual production, process planning and scheduling are closely linked. But, in traditional production, process planning and scheduling are often carried out separately, resulting in mutual constraints. The main performance is that process planning often causes conflicts and unreasonable allocation of production resources.

Henan Provincial Key Laboratory of Intelligent Manufacturing of Mechanical Equipment, Zhengzhou University of Light Industry, Zhengzhou, China

Corresponding author:
Xiaoyu Wen, Henan Provincial Key Laboratory of Intelligent Manufacturing of Mechanical Equipment, Zhengzhou University of Light Industry, No.5 Dongfeng Road, Zhengzhou, Henan 450002, China.
Email: wenxiaoyu@zzuli.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
Therefore, it is difficult to reach the best scheduling plan. Meanwhile, due to the influence of many uncertain factors in the scheduling process, the process plans often need to be modified. In the 1980s, some scholars proposed the integrated process planning and scheduling (IPPS) problem. After decades of development, solving IPPS has accumulated abundant theoretical and practical achievements.

Many results of researches show that IPPS is of great significance to resolution resource conflicts, improvement of equipment utilization ratio and production efficiency. The research of IPPS has become an effective way to solve the problems of workshop production management. IPPS is an expansion of shop scheduling problem, so the problem solution space is larger and more complicated. Process plan flexibilities, machine flexibilities, different jobs, different operations, and various resources and operations’ constraints should be considered simultaneously for solving IPPS problems. The IPPS problem is non-deterministic polynomial (NP)-complete problem. With the development of new information technology, many intelligent optimization algorithms are widely used in intelligent manufacturing. A large number of swarm intelligent algorithms are utilized in IPPS, which provide a theoretical guidance for the realization of intelligent manufacturing.

With the in-depth study of related IPPS, new problems are constantly being discovered. Workshop production is a complex process, so solving IPPS problem can not only meet a single objective, but need to consider multiple objectives. Generally, it is necessary to consider production efficiency, due date and other objectives. In the production process of the workshop, tool loading and unloading, the proficiency of personnel in the workshop, the operating status of the equipment, the transmission status of the jobs are usually uncertain conditions. Therefore, considering the uncertain factors in the IPPS problems is more in line with the actual production status. Especially, with the requirements of on-time and the development of intelligent manufacturing technology, it is vital to fully estimate the system uncertainty. Considering multi-objective and uncertain factors is helpful to make the research closer to the actual production situation. But it also makes the IPPS problem more complicated. Therefore, it is a significant work to design an appropriate model and an effective solution method.

At present, there are many studies on multi-objective IPPS and uncertain IPPS. The weighted method and Pareto method are adopted to solve the multi-objective problem in the existing research. The uncertainty of the system is mainly expressed by fuzzy numbers, interval numbers, and random numbers, such as fuzzy due date, fuzzy processing time, etc. It noticed that there are few researches IPPS considering uncertainty and multi-objectives at the same time. Therefore, in this paper, an effective layered algorithm is designed for the fuzzy multi-objective IPPS (FMOIPPS) problem. Process planning and workshop scheduling are both very complex problems. The IPPS problem is extended on the basis of process planning and scheduling, which makes the problem solving more complex. The FMOIPPS problem considers the uncertain factors and multi-objective of the production process on the basis of the IPPS problem, making the solution of the problem more difficult. Therefore, a multi-objective collaborative optimization (MLCO) method is designed to solve the problem in three steps, and a better solution is finally obtained through three levels of cooperation. The MLCO method including three layers of process planning layers, process selection layers and scheduling layers is designed. The basic genetic algorithm is utilized in the process planning layer to produce the process plans for jobs. In the process selection layer, considering the processing information of all jobs, a multi-objective genetic algorithm (MOGA) is designed to select the process plan for each job. In the scheduling layer, a MOGA with a boundary search strategy is proposed to assign process information for all jobs. For the multi-objective problem, the Pareto-based method is used, and the final result is a Pareto solution set. Finally, 32 sets problem examples of different scales are employed to test feasibility and effectiveness of the MLCO method.

The remainder of this paper is organized as follows: “Literature Review” section introduce the development of the IPPS problem. “Problem description” section elaborates the FMOIPPS problem settled in this paper. “The proposed MLCO method” section presents proposed MLCO method for fuzzy multi-objective IPPS. “Experiments and discussions” section, the experiments design and result analyses are given. The conclusions and future works are reported in last section.

**Literature review**

IPPS was first proposed by Chryssolouris et al. After decades of development, IPPS has accumulated vast theoretical and practical values. Li et al. summarized the research results of IPPS problem and proposed the future development direction. At present, the research on IPPS mainly focuses on model establishment and solution method.

There are three types of IPPS problem modeling, non-linear process planning, closed loop process planning, distributed process planning. Many models for IPPS problems have been established. Jain et al. adopted a two-module integration method to select the process plan through process planning and scheduling. Li et al. designed an agent-based IPPS integration method, which process planning and scheduling are executed simultaneously and use optimized agents based on evolutionary algorithms to manage the interaction and communication between agents. Phanden et al. proposed the integration of four modules: process plan selection module, scheduling module,
schedule analysis module, and plan modification module. Chu et al.\textsuperscript{23} proposed an integrated model of uncertain bi-level programming. The upper layer solves planning problem and the lower layer solves scheduling problems.

Over the years, the algorithm for solving IPPS problem has developed rapidly. For single-objective optimization, Seker et al.\textsuperscript{24} designed a hybrid algorithm by combining GA and fuzzy neural network for the IPPS problem with makespan as the optimization objective. Li et al.\textsuperscript{25} proposed an active learning genetic algorithm with makespan as the objective to solve the IPPS problem. Leung et al.\textsuperscript{26} used makespan as the optimization objective, and proposed an ant colony optimization algorithm based on agent system to solve the IPPS problem. Zhang et al.\textsuperscript{27} designed an objective-coding genetic algorithm solution with makespan as the optimization objective and achieved well results. Sobeyko et al.\textsuperscript{28} adopted a heuristic algorithm based on variable neighborhood search with the goal of total weighted tardiness. The single-objective optimization algorithm has abundant accumulation, and the algorithm mainly used makespan as the optimization objective. Recently, the use of multi-objective algorithm to solve IPPS is increasing. Shokouhi et al.\textsuperscript{29} constructed a mathematical model considering multiple factors to solve the complexity of IPPS and proposed a multi-objective genetic algorithm with three objectives: makespan, tardiness, and machine workload and machines total workload. Li et al.\textsuperscript{30} designed a novel hybrid algorithm for the multi-objective IPPS problem based on Nash equilibrium in game theory. Luo et al.\textsuperscript{31} proposed a multi-objective genetic algorithm based on the principle of immunity and external archive for the MOIPPS problem.

Due to the consideration of multi-objective to meet the actual production needs of the workshop, multi-objective algorithm has been widely used in solving the workshop scheduling problem. Li et al.\textsuperscript{32} studied the process and scheduling problems of welding workshop. Aiming at the WSSP problem in real life, a multi-objective mathematical model considering both energy efficiency and production efficiency is proposed, and an effective multi-objective artificial bee colony algorithm is designed to solve the problem. Yin et al.\textsuperscript{33} established a low-carbon scheduling model for flexible job shop scheduling problem, considering the three objectives of productivity, energy efficiency and noise reduction, and proposed a multi-objective genetic algorithm. Lu et al.\textsuperscript{34} proposed a model of multi-objective dynamic welding scheduling problem and designed a multi-objective gray wolf optimization algorithm to solve the problem. Lu et al.\textsuperscript{35} designed a multi-objective mathematical model considering both the maximum makespan and energy consumption for the flow shop scheduling problem, and adopted a hybrid multi-objective backtracking search algorithm. Consider multi-objective to make the production plan more in line with the requirements. So, the design of multi-objective algorithm is of great significance for the solution of practical problems.

Recently, uncertain factors have been extensively considered in various scheduling problems. Gao et al.\textsuperscript{36} studied the scheduling problem of flexible workshops, established an FJSP model with uncertain processing time, the uncertainty of processing time is represented by triangular fuzzy number. Jamrus et al.\textsuperscript{37} considering the uncertainty of process and equipment control process, fuzzy number is adopted to express the uncertainty of processing time, used a hybrid algorithm combining particle swarm optimization and genetic operators for flexible job shop scheduling problems. Lei\textsuperscript{38} applied interval number theory to scheduling, established the scheduling model of interval number processing time, and it is proved by examples that the real makespan of each scheme is within the makespan interval indicated by the interval number. Lei\textsuperscript{39} established the mathematical model of FJSP problem with fuzzy processing time, and aiming at the complexity of FJSP problem, an effective co-evolution algorithm is proposed. Hu et al.\textsuperscript{40} proposed an uncertain job shop scheduling problem with fuzzy numbers, which represents the processing time and due date, and established a corresponding objective function to ensure that the optimal scheduling plan is obtained. According to the uncertain characteristics of the processing time in the remanufacturing workshop, Gao et al.\textsuperscript{41} use the fuzzy number to express the processing time and an effective discrete harmony search algorithm is designed to solve it. Gao et al.\textsuperscript{42} took the fuzzy processing time into consideration in flexible job shop scheduling, use the maximum fuzzy makespan as the objective, and an improved two-stage artificial bee colony algorithm is designed for FJSP with fuzzy processing time. Lu et al.\textsuperscript{43} considering the uncertainty of processing parameters, studied the multi-objective FJSP with CPT problem and established the corresponding mathematical model. Finally, a new multi-objective discrete virus optimization algorithm is designed to solve this problem.

Gradually, uncertain factors have also been taken into account in the IPPS problem by some scholars. Zhang et al.\textsuperscript{15} adopted triangular fuzzy numbers to express the processing time and transportation time of the machine, and solved distributed integration of fuzzy process planning and scheduling. Li et al.\textsuperscript{16} used interval numbers to represent the processing time, proposed an uncertain IPPS model, and designed a corresponding hybrid algorithm combining genetic algorithm and particle swarm algorithm. Wen et al.\textsuperscript{17} designed a multi-objective IPPS mathematical model with uncertain processing time and fuzzy due date, and designed an improved bee colony algorithm for the problem. Haddadzade et al.\textsuperscript{2} used random numbers to represent the processing time, and designed a hybrid algorithm combining simulated annealing and tabu search. The
research results show that taking multi-objective and uncertainty factors into account in the IPPS problem increases the difficulty of solving, but makes the problem closer to the actual situation, which is conducive to improving the quality of the scheme. It is very complicated for solving IPPS problems considering both uncertainty and multi-objectives, therefore, this paper design an effective method to solve FMOIPPS problems.

**Problem description**

The research of fuzzy set theory to deal with uncertain parameters of scheduling has gained wide attention in the field of scheduling. Trapezoidal fuzzy number and triangular fuzzy number are used to express the uncertainty of due date and processing time, respectively. The IPPS problem belongs to the field of shop scheduling problems, so this paper also applies fuzzy set theory to solve the FMOIPPS problem. In mathematical model, triangular fuzzy number and trapezoidal fuzzy number are used to represent fuzzy processing time and fuzzy due date respectively, and the fuzzy numbers operations are presented from Wen et al.17

The FMOIPPS problem studied in this paper can be described as follows: There is a set of jobs $J = \{J_1, J_2, J_3, \ldots, J_n\}$ to be processed in a factory equipped with a set of machines $M = \{M_1, M_2, M_3, \ldots, M_m\}$. Each job has various machining features which satisfying certain precedence constraints. There are many optional operations for each feature of a job. Each process has multiple processing machines with precedence constraints among operations, however, the processing time and due date of each job are fuzzy numbers. Solving the FIPPS problem is to find a processing plan that satisfies multiple objectives at the same time based on the given resources and processing information.

The mathematical model of FMOIPPS and the fuzzy numbers operations can be found in Wen et al.17 In order to better describe the mathematical model, first define the following symbols:

- $n$: the number of all jobs
- $J_i$: the set of jobs.
- $\tilde{C}$: the fuzzy makespan of $J_i$
- $ST$: the average customer satisfaction
- $MS$: fuzzy the spread of fuzzy makespan
- $C_1$: the best completion time of job
- $C_3$: the worst completion time of job

Fuzzy maximum makespan, fuzzy average customer satisfaction and the spread of fuzzy makespan are used as the optimization objectives for solving the FMOIPPS in this paper, the objective functions are given as follows:

$f_1$: Minimizing fuzzy makespan

$$\text{Min } f_1 = \tilde{C} = \max_{i \in [1, n]} \tilde{C}$$

$f_2$: Minimizing average customer satisfaction

$$\text{Max } f_2 = ST = \frac{1}{n} \sum_{i=1}^{n} ST_i$$

$f_3$: Minimizing the spread of fuzzy makespan

$$\text{Min } f_3 = MS = C_3 - C_1$$

In the above FMOIPPS problem model, the smaller the fuzzy makespan, the smaller costs and energy. The greater the average customer satisfaction, the greater the probability that jobs of on-time delivery. The evaluation method of customer satisfaction and the calculation method of average customer satisfaction were proposed by Wen et al.17 The shorter the span of the fuzzy makespan, which is conducive to cost savings, reduced energy consumption, and improved on-time of completion time of jobs.

The following constraints also need to be taken into account:

1. Only one operation for each job must be processed simultaneously.
2. Only one operation can only be processed on each machine at a time.
3. Each job can only choose one process plan at a time.
4. Only one optional processing machine for each operation can be chosen.
5. The processing process is not limited by the buffer.

A simple example is given to better clarify the FMOIPPS problem in Table 1. Each job has different features with precedence constraints, some features have multiple operations companying operations sequences, each operation can be machined various machines with different fuzzy processing time, each job has different fuzzy due date. In Table 1, Job 1 includes 6 features, and the first feature $F_1$ must be machined before all features. The first and second features of job 1 have 2 alternative operations. The operations sequence of first candidate operations for the first feature $F_1$ have two operations $O_1$-$O_2$, however, the operations sequence of second candidate operations of $F_1$ contains only one operation $O_3$. There are various machines to process some operations. $O_1$ can be processed on two machines, which are $M_1$ and $M_2$. Moreover, different processing machines have different fuzzy processing times, and different processed jobs contain different fuzzy due date.

According to the processing information in Table 1, the processing time of the first process $O_1$ of the $J_1$ on the machine $M_1$ is $(7, 8, 9)$, which means that the shortest processing time of $O_1$ is 7, the longest processing time is 9, and the processing time with the greatest
possibility is 8. The due time of $J_1$ is (66, 86, 126, 146), which means that the earliest due time of the job is 66, the latest due time is 146, and the best due time is 86 to 126. When a scheduling plan is determined, the processing technology, process sequence and processing machine of each process will be determined. Through the fuzzy number summing operation, the fuzzy makespan of each job can be obtained by the operation of fuzzy number summation, and then calculate the value of the three objective functions. Get the final solution through comparison.

**Proposed multi-layer collaborative optimization method for fuzzy multi-objective IPPS**

**Flowchart of MLCO**

MLCO method is designed to address the FMOIPPS problem, including three layers. For the process planning layer, the basic genetic algorithm (GA) is applied to provide various near optimal process plans for the process selection system. For the process selection layer, a multi-objective genetic algorithm (MOGA) is designed to optimize the process selection population and select a process plan for each job, considering the overall machining process of all jobs. An individual comprehensive evaluation method is introduced to evaluate non-dominated solutions. A sharing function method is introduced to maintain population diversity. Based on the NSGAII algorithm, the basic elements of fast non-dominated sorting, elite strategy, crowded distance, and tournament selection is used in the optimization method. The external archive method is adopted to preserve the non-dominated solutions generated during population evolution. Minimizing processing time of all jobs and processing time with the maximum processing machine are selected as the optimization objectives. The goal of process selection layer is to combine optimization of the process plans of all jobs. For the scheduling layer, a MOGA with a boundary search strategy is adopted. The boundary search strategy is

| Job | Features | Candidate operations | Candidate machines | Precedence constraints | Fuzzy process time | Fuzzy due date |
|-----|----------|----------------------|--------------------|-----------------------|-------------------|----------------|
| J1  |          | O₁-O₂                | M₁, M₂, M₈         | Before All            | (7.8,9,5.6,7)    | (66,86,126,146) |
|     |          | O₃                   | M₅                 |                       | (11,14,16)       | (6,7,8)        |
| F₂  |          | O₄                   | M₆                 |                       | (7,10,12)        | (9,13,15)      |
|     |          | O₃-O₆               | M₁, M₂, M₃, M₆    |                       | (8,10,11)        | (9,13,15)      |
|     |          | O₇                   | M₉                 |                       | (6.7,9)          | (5.8,9)        |
| F₄  |          | O₈-O₉               | M₁₀                |                       | (5.7,10)         |               |
|     |          | O₁₀                  | M₁₀, M₂, M₃       |                       | (11,14,16)       | (10,15,17)     |
| F₅  |          | O₁₁                  | M₃                 |                       | (10,13,16)       |               |
|     |          | O₁₂                  | M₁, M₂, M₃        |                       | (6,10,13)        | (7,10,13)      |
| J₂  |          | O₁                   | M₄, M₅             | Before All            | (5.8,10)         | (24,44,84,104) |
|     |          | O₂                   | M₆, M₈             |                       | (6,8,10)         | (8,10,11)      |
| F₂  |          | O₃-O₄               | M₉                 |                       | (8,10,12)        | (7,9,11)       |
|     |          | O₅                   | M₉                 |                       | (8,9,12)         |               |
| F₃  |          | O₃-O₆               | M₉                 |                       | (8,10,12)        | (7,9,11)       |
|     |          | O₇-O₈               | M₉                 |                       | (10,13,16)       |               |
|     |          | O₁₀-O₁₁             | M₉                 |                       | (9,12,14)        |               |
| F₄  |          | O₉                   | M₁, M₂, M₂         | Before F₇             | (12,14,16)       | (15,18,20)     |
|     |          | O₁₀-O₁₁             | M₁, M₆             |                       | (7,9,12)         | (14,16,19)     |
|     |          | O₁₃-O₁₄             | M₉                 |                       | (6,8,11)         |               |
| F₅  |          | O₁₂                  | M₁, M₂, M₂         | Before F₆             | (18,20,21)       | (15,17,22)     |
|     |          | O₁₃-O₁₄             | M₉                 |                       | (8,10,12)        | (16,18,23)     |
|     |          | O₁₃-O₁₄             | M₂, M₇             |                       | (8,10,12)        | (7,10,13)     |
| F₆  |          | O₁₅                  | M₂, M₇             |                       | (8,11,13)        |               |
| F₇  |          | O₁₆                  | M₁, M₂, M₂         |                       | (10,13,15)       | (15,16,18)     |
|     |          | O₁₇-O₁₈             | M₂, M₇             |                       | (8,10,12)        | (7,11,14)     |
|     |          | O₁₉                  | M₁, M₂, M₃         |                       | (8,10,12)        |               |
| F₇  |          | O₂₀-O₂₁             | M₉                 |                       | (20,22,25)       | (20,22,24)     |
|     |          | O₂₀-O₂₁             | M₈                 |                       | (7,9,12)         | (8,10,12)     |
|     |          | O₂₀-O₂₁             | M₁₀                |                       | (12,14,16)       |               |
designed to improve the search ability of boundary solutions. Three optimization objectives are minimizing fuzzy makespan, maximizing average customer satisfaction and minimizing the spread of fuzzy makespan simultaneously. The target of scheduling layer is to make scheduling arrangements for the process information obtained by process selection layer. Through mutual cooperation among each layer, guide the overall optimization process, and finally get satisfactory solutions. The workflow of proposed MLCO method for FMOIPPS is shown in Figure 1.

The main steps of MLCO method are as follows:

Step 1: Initialize process planning population.
Step 2: The optimized process plans is provided for each job through GA.
Step 3: Generate process selection population based on process plans’ information of jobs.
Step 4: Use MOGA to optimize the process selection population and select a process plan for each job, considering the overall machining process of all jobs.
Step 5: Initialize the scheduling population based on the processing information of jobs.
Step 6: Use a MOGA with a boundary search strategy to optimize scheduling population and generate the scheduling scheme.
Step 7: If the number of current generation reached, output the optimized solution. Otherwise, get on first step.

Methods for process planning layer

For the process planning layer, the basic GA is adopted to optimize the completion of each job. The process planning layer mainly provides the near optimal process plans for the process selection layer. Minimizing fuzzy processing time is selected as the optimization objectives. The workflow of GA for process planning layer is shown in Figure 2.

The detailed steps of population optimization in process planning layer as follows:

Step 1: Randomly initialize population for each job.
Step 2: Calculate the fitness value of each individual. Take the fuzzy processing time of job as individual fitness value.
Step 3: According to the fitness value, tournament selection is adopted to select parent individuals. The crossover operators and mutation operators proposed by Li et al. and were adopt to generate new population.

Step 4: Offspring selection: According to the comparison criteria of triangular fuzzy numbers, selecting individuals with small fuzzy processing time to generate new population \( P_1 \).

Step 5: When evolutionary algebra is satisfied, go to Step 6. Otherwise, go to Step 2.

Step 6: Select near optimal process plans and input them to process selection layer.

Step 7: From Step 2 to Step 6 is repeat to complete the optimization of all jobs populations.

Encoding and decoding: The multi-dimensional encoding method and decoding method from Li et al. is adopted for process planning populations. An individual consists of three chromosomes, there is a feasible encoding scheme is constructed in Figure 3. There are 3 chromosomes in total, the length of the features string in chromosome 1 and the alternative operations string in chromosome 2 is the number of the features. For the features string, the features string contains 4 features, the features sequence of first part in chromosome 1 is \( F_2 \rightarrow F_3 \rightarrow F_1 \rightarrow F_4 \). For the alternative operations string, the element 3 at the fourth position shows that the process 3 is selected as the candidate operation of the feature \( F_4 \). For the alternative machines string, the element 4 at the fifth position represents that operation \( O_3 \) chooses the candidate processed machine \( M_4 \). The length of the alternative machines string is the total number of all operations. The decoding method is presented in Li et al.

Methods for process selection layer

The process selection is mainly responsible for providing process information for the scheduling system. The process planning population only generates process plan of a single job, but the scheduling is for all jobs. Therefore, it is important to consider the processing plan of each job base on processing information of all the jobs. The main purpose of this layer is to determine the processing information of all jobs. MOGA is designed to optimize the process selection population and select a process plan of each job based on the two indexes of the minimizing processing time of all jobs and processing time with the maximum processing machine. The MOGA algorithm adopts fast non-dominated sorting based on NSGAII, individual crowding strategy and elite retention strategy. The Pareto-based method is used to deal with multi-objective problems. Introduce the non-dominated solution obtained from the optimization process of external archive. In addition, the method of sharing function is used to reduce the individual fitness value in the solution dense area, reduce the possibility of producing offspring, and maintain the diversity of the population. A method of individual comprehensive evaluation is designed to select the branch solution to the external archive or to the new generation population. The workflow of MOGA for process selection layer is shown in Figure 4.

The detailed steps of population optimization in process selection population optimization layer are presented as follows:

Step 1: According to the information provided by process planning, initialize population \( P_2 \) randomly.

Step 2: Calculate the fitness value of individuals of \( P_2 \) population.

---

**Figure 2.** The workflow of GA for process planning layer.

**Figure 3.** Encoding of process planning layer.
Step 3: Processing time of all jobs and the processing time with the maximum processing machine as evaluation indicators. The Pareto method and the triangular fuzzy number comparison method, non-dominated rank sorting is used to allocate dominant ranks to individuals in the population $P_2$. Then update the fitness value of the individual by the sharing function.

Step 4: According to the two indexes of dominance level and population fitness value, the individual is selected as the parent from the population by using tournament selection, and the crossover and mutation operators designed by the alternative operation string are used to generate new populations.

Step 5: Process selection population $P_2$ and population $P_{new}$ into population $P_{merge}$, and assign dominant levels to populations. Calculate the individual crowding degree, and then calculate the individual’s comprehensive score according to the individual comprehensive evaluation method.

Step 6: The external archive 1 is updated with the non-dominated solution of the population $P_{merge}$ and the individual’s comprehensive scores. If the number of non-dominated solutions is larger than the size of external archive, the individual with high comprehensive score should be selected.

Step 7: Generate a new generation of population. Firstly, the non-dominated solution is liberated into a new population, and then the new population is filled according to individual crowding degree. If the number of non-dominated solutions is greater than the population $P_2$, it will be selected based on the overall score.

Step 8: If the evolutionary generation is satisfied, go to Step 9. Otherwise go to Step 2.

Step 9: Randomly select an individual from the non-dominated solution of the archive 1, input the revised process plan into the scheduling system.

Encoding and decoding scheme. The encoding scheme is shown in the Figure 5. Each individual contains a chromosome. The length of the chromosome is the number of all jobs. The element 3 in the first position indicates that the job 1 selects the process plan P3. From the process selection string, the process plans selected for the 4 jobs can be decoded as $P_3\rightarrow P_2\rightarrow P_5\rightarrow P_4$. The process
Table 2. The ranking of individual individuals under different objectives.

| Individual | Ranking of objective 1 | Ranking of objective 2 | Ranking of objective 3 | Comprehensive scores |
|------------|------------------------|------------------------|------------------------|----------------------|
| \( \hat{a} \) (6,4,46) | 1 | 1 | 5 | 17 |
| \( b \) (9,57,23) | 2 | 2 | 1 | 16 |
| \( c \) (11,75,39) | 3 | 4 | 3 | 8 |
| \( d \) (16,64,31) | 4 | 3 | 2 | 9 |
| \( e \) (51,80,45) | 5 | 5 | 4 | 4 |

Step 4: Update the fitness value

Step 3: Compare the regional range \( \text{vidual} \ k \).

For individual \( \text{vidual} \ i \), the minimum value found, the average value of fuzzy number is used. \( FV \) and \( SV \) are weight coefficients, the value of both is 0.5.

Step 5: Repeat Step 2 and Step 4, and complete the comparison between individual \( i \) and all individuals in the population.

Step 6: Repeat the above steps to update the fitness value of all individuals in the population.

**Fitness evaluation.** Considering the processing time of all jobs and processing time with the maximum processing machine, the individual fitness value is calculated as follows:

\[
F(x) = \omega_1 \frac{f_i(x) - f_{\min}}{f_{\max} - f_{\min}} + \omega_2 \frac{s_i(x) - s_{\min}}{s_{\max} - s_{\min}}
\]

(4)

In the formula (4), \( f_i(x) \) is the processing time of all jobs, and \( s_i(x) \) is the processing time with the maximum processing machine, \( t_{\max} \) and \( s_{\max} \) represent the maximum value found, \( t_{\min} \) and \( s_{\min} \) represent the minimum value found, the average value of fuzzy number is used. \( \omega_1 \) and \( \omega_2 \) are weight coefficients, the value of both is 0.5.

**Sharing function.** The sharing function is a method to determine the decline in the fitness value of an individual according to how close the individual is to other individuals within a certain distance. As the fitness value decreases, individuals in densely populated areas are less likely to generate offspring. The detailed steps of the sharing function are described as follows:

Step 1: Select individual \( i \) in the population and calculate the regional range \( \text{Zij} = f_i \times rc \) of each objective for individual \( i \), \( f_i \) is the \( j \)th objective function value of individual \( i \), range factor \( rc \) is equal to 0.02.

Step 2: Calculate the difference \( |\Delta_j| \) between each objective function value of individual \( i \) and individual \( k \).

Step 3: Compare the regional range \( \text{Zij} = f_i \times rc \) of each objective value and \( |\Delta_j| \), and use variable \( n \) to record the number where \( \text{Zij} \) is greater than \( |\Delta_j| \).

Step 4: Update the fitness value \( f_{ij} = f_{ij} \times n \times ac \) of individual \( i \), \( f'_{ij} \) is the individual fitness value of the previous state, fitness value attenuation coefficient \( ac \) is equal to 0.98.

**Individual comprehensive evaluation.** Multi-objective optimization is usually to obtain a solution with better overall performance. In the population, individuals with better overall performance usually carry more valuable information. A comprehensive performance evaluation method based on ranking individuals is designed in this paper. According to the ranking of individuals, the problem of dimensional non-uniformity between the objective values can be ignored, and the impact of each objective on the individual performance can be considered. At the same time, the value of a single objective excellent individual is considered, and the weight of the first ranked individual need to be increased.

Suppose a population size \( n \) is 5, and the number \( m \) of optimization objective is 3. The individual populations are represented by vectors \( \hat{a}(6, 4, 46) \), and the numbers represent the function values of the three optimization objectives. According to the function value of each objective, the individuals of the population are sorted separately, and the ranking of the individual under each objective can be obtained, as shown in Table 2. Calculate the comprehensive score of each individual \( C = m \times (n + 1) - (r_1 + r_2 + r_3) + Z \times W \times (n + 1) \), where \( m \) is the number of objective, \( n \) is the population size, \( r_i \) is ranking under \( ith \) objective of the individual, \( Z \) is the number of times the individual ranks first, the weight coefficient \( W \) is 0.5. The individual’s comprehensive scores are shown in Table 2.

Comprehensive performance is used to evaluate an individual’s overall performance. A higher score indicates that the individual’s overall performance is better. Conversely, if the score is lower, the individual’s overall performance is worse.

**Update external archive.** In this paper, the external archive method is adopted to preserve the non-dominated solutions generated during population
The solution to update the external archive can be a non-dominated population solution or other archive solutions. The external archive is updated as follows:

**Step 1:** Combine the non-dominated solution of the population, or the solution of other archive with the solution of the current archive to be combined into a combined solution set $S$.

**Step 2:** According to the function value of the optimization objective for the population, according to Pareto definition and comparison method of triangular fuzzy number, fast non-dominated sorting is used to find the non-dominated solutions in the merged solution set $S$.

**Step 3:** Empty the current archive, liberate the non-dominated in the merged solution set $S$ into the current archive.

**Methods for job shop scheduling layer**

The optimization of scheduling population is the most important part of the overall optimization process. It mainly completes the arrangement of processing sequence and processing time of jobs. A MOGA with a boundary search strategy is adopted. The objectives of scheduling optimization for FMOIPPS are minimizing fuzzy makespan, shorting spread of fuzzy makespan and maximizing customer satisfaction simultaneously. The algorithm used in the scheduling layer is based on the MOGA algorithm with a boundary search strategy. Boundary search is to perform genetic operations on non-dominated solutions generated in the optimization process to enhance the search capabilities of boundary solutions. The workflow of MOGA with a boundary search strategy for scheduling layer is shown in Figure 6.

The detailed steps of population optimization in scheduling layer are presented as follows:
Step 1: Initialize the scheduling population $P_3$ according to the process information provided by the process selection layer.

Step 2: Calculate the fitness value of population $P_3$.

Step 3: Fuzzy makespan, spread of fuzzy makespan and average customer satisfaction are optimization indicators. Decode processing information of each individual, and solve the three objective values of the individual. According to the Pareto method and the triangular fuzzy number comparison method, non-dominated rank sorting is used to allocate dominant ranks to individuals in the population $P_3$. Then update the fitness value of the population $P_3$ by the sharing function.

Step 4: According to the two indexes of dominance rank and population fitness value, the individual is selected as the parent from the population by using the tournament method. A new population is generated by adopt crossover operation and mutation operation proposed by Li et al.8

Step 5: Individual selection: merge population $P_3$ and population $P_{new}$ into population $P_{merge}$, and assign dominant ranks for population $P_{merge}$. Calculate the individual’s crowding degree, and then calculate the individual’s comprehensive score according to the individual’s comprehensive evaluation method. Update the archive 3 with the non-dominated solution in the population $P_3$ by the sharing function.

Step 6: If there is a solution in population $P_{merge}$ that dominates the solution in external archive 2, then update the external archive 2 with the non-dominated solution in the population $P_{merge}$, and then go to Step 7. Otherwise, go to Step 8.

Step 7: Perform the boundary search.

Step 7.1: Each non-dominated solution from the archive 2 is randomly crossed with other non-dominated solutions according to POX operator or a crossover operator based on job to generate offspring, and these individuals and individuals in the external archive 2 are formed into boundary population $P_4$. Clear the individuals in external archive 2.

Step 7.2: Calculate the fitness value of population $P_4$.

Step 7.3: According to the method of scheduling optimization in Step 3, calculate the dominance level and update fitness value of the population $P_4$.

Step 7.4: According to the method of scheduling optimization in Step 4, generate population $P_{new}$.

Step 7.5: Assign dominant ranks for population $P_{merge}$, calculate the individual’s crowding degree, and then calculate the individual’s comprehensive score according to the individual’s comprehensive evaluation method. Update the external archive 3 with non-dominated solutions in the population $P_{new}$. If the number of non-dominated solutions is greater than the size of the archive, then update the archive according to the individual’s comprehensive scores.

Step 7.6: If the evolution times are satisfied, then go to Step 7.8. Otherwise, carry on next step.

Step 7.7: According to the method of scheduling optimization in Step 8, generate a new generation of population $P_3$ and go to Step 7.2.

Step 7.8: Combine the population $P_{merge}$ and the scheduling population $P_{new}$ into a population $P_{merge}$, assign a dominance level to the population, calculate the individual crowding degree, place the individual into population $P_{merge}$ according to the two indicators, and go to Step 9.

Step 8: update the external archive 2 with the non-dominated solution in the population $P_{merge}$.

Step 9: Liberate the non-dominated population from the population $P_{merge}$ into a new population, and then fill the new population according to the crowding degree. If the number of non-dominated solutions is greater than the population size, select individuals to generate a new population based on comprehensive scores.

Step 10: If the evolution times are satisfied, go to last step. Otherwise, go to Step 3.

Step 11: Output the solution of external archive 3.

**Encoding and decoding scheme.** Operation-based encoding method is used in this paper. As seen in Figure 7, the length of the scheduling string is the number of all the operations for jobs. The element in the scheduling string represents the jobs’ serial number. The element 2 in the first position indicates the first operation of job 2 and the number 2 in the 3rd position denotes the second operation of job 2. The greedy decoding method is adopted to decode the coding scheme into active scheduling, which can get the start time, end time of each job.

**Fitness evaluation.** Three objective functions of minimizing fuzzy makespan $f_1$, maximizing average customer satisfaction $f_2$, and minimizing the spread of fuzzy makespan $f_3$ are selected for scheduling layer. Considering the conflict between objectives, the individual fitness value is replaced by the individual comprehensive evaluation score.
**Boundary search**

The global optimal solution of many optimization problems usually exists on the boundary of the feasible region, especially the non-dominated solution generated by the evolution of the population, which often carries the information closest to the optimal solution in the current population. It is very necessary to mine the information of these solutions comprehensively. So the non-dominated solutions generated during the evolution of the scheduling population are randomly combined to generate the boundary search population and perform the genetic search in this paper. The operation flow of boundary search is as follows:

**Step 1:** The obtained non-dominated solutions are sequentially cross-operated to generate the initial boundary population.

**Step 2:** Calculate individual fitness value.

**Step 3:** Genetic operation to generate new individuals.

**Step 4:** Select individuals to form a new generation of population.

**Step 5:** If the evolutionary algebra is satisfied, go to step 6, otherwise return to step 2.

**Step 6:** Merging boundary populations and scheduling populations.

**Experiments and discussions**

**Experiments**

Due to lack of FMOIPPS benchmarks, 32 sets of examples with different sizes are designed to verify the proposed MLCO method for solving the FMOIPPS problems in Table 3. The processing information of an instance with 5 jobs processed in 5 machines is shown in Table 4.

MLCO, NSGAII and NSGAII + N5 three methods are used to solve 32 sets of cases with various scales. NSGAII and NSGAII + N5 as two control groups, adopting two-stage optimization methods, namely the process planning stage and the scheduling stage. The process planning stage provides near-optimal process plans for each job, and the scheduling stage arranges the processing sequence and processing time of the jobs. GA is used in the process planning stage, and the NSGAII algorithm, NSGAII algorithm with N5 neighborhood search algorithm are employed in the scheduling stage. The performance of the algorithms is evaluated from the convergence and divergence of the algorithm and the ability to find solutions.

The operating environment is as follows: the three algorithms is programmed by Visual C+++. The computer performance is intel (R)Core(TM)i5-5000 CPU@2.00GHz. The parameters in the proposed algorithm are selected after a lot of trials. The selected parameters are listed in Table 5.

**Evaluation indicators of algorithm performance**

**Generation distance.** Generation distance (GD) is used to express the convergence of the obtained solution set. The smaller the GD value, the closer the solution set obtained by the algorithm to the real Pareto optimal solution set. The GD indicator represents the distance between the known Pareto front end ($PF_{known}$) and the real Pareto front end ($PF_{true}$), and its calculation formula is as follows:

$$GD = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}}$$  \hspace{1cm} (5)

In the formula (5), $n$ is the number of solutions in $PF_{known}$, $d_i$ is the distance between the $ith$ point in $PF_{known}$ and the nearest point in $PF_{true}$.

**Inverse generation distance.** Inverse generation distance (IGD) is used to simultaneously evaluate the convergence and diversity of the solution set. The smaller the IGD value, the better the algorithm performance. IGD is distance $PF_{known}$ and $PF_{true}$. The distance is calculated as follows:

$$IGD = \sqrt{\frac{\sum_{i=1}^{m} d_i^2}{m}}$$  \hspace{1cm} (6)

In the formula (6), $m$ is the number of solutions in $PF_{true}$, $d_i$ is the distance between the $ith$ point in $PF_{known}$ and the nearest point in $PF_{true}$.

---

**Table 3.** 32 sets examples of different scale.

| Cases | Numbers of machine | Numbers of job | Cases | Numbers of machine | Numbers of job | Cases | Numbers of machine | Numbers of job | Cases | Numbers of machine | Numbers of job |
|-------|--------------------|----------------|-------|--------------------|----------------|-------|--------------------|----------------|-------|--------------------|----------------|
| 1     | 5                  | 5             | 2     | 5                  | 10            | 3     | 5                  | 15            | 4     | 5                  | 20            |
| 1     |                    |               | 1     | 10                 | 8             | 5     | 5                  | 7             | 6     | 7                  | 10            |
| 2     | 5                  | 10            | 1     | 15                 | 8             | 6     |                    | 14            | 7     |                    | 16            |
| 2     |                    |               | 2     | 20                 | 12            | 7     |                    | 13            | 8     |                    | 10            |
| 3     |                    |               | 3     |                    | 10            | 8     |                    | 15            | 9     |                    | 25            |
| 3     |                    |               | 4     |                    | 12            | 9     |                    | 8             | 10    |                    | 20            |
| 4     |                    |               | 5     |                    | 13            | 10    |                    | 14            | 11    |                    | 25            |
| 5     |                    |               | 6     |                    | 15            | 12    |                    | 15            | 13    |                    | 30            |
| 6     |                    |               | 7     |                    | 14            | 15    |                    | 17            | 14    |                    | 35            |
| 7     |                    |               | 8     |                    | 16            | 17    |                    | 18            | 15    |                    | 40            |

---

---

Measurement and Control 53(9-10)

---
For each group of cases, three algorithms are used to run independently 20 times. The solutions obtained by each algorithm constitute their own PFknown, and the un-dominated solution set obtained by combining the solutions of all algorithms constitutes PFtrue. Due to the lack of reference points, the non-dominated solution set obtained by combining the non-dominated solutions found by the three algorithms is used as the reference point. Table 6 gives the reference points for experiment 1. The calculated results about average value of GD and IGD are shown in Table 7.

According to the result in Table 7, the GD values obtained using the MLCO method are all less than the GD values obtained by NSGAII, NSGAII + N5, except for the 31 sets of experiments, the GD of other groups are less than 1 and mostly maintained at below

| Job | Features | Candidate operations | Candidate machines | Precedence constraints | Fuzzy process Time | Fuzzy due date |
|-----|----------|----------------------|---------------------|-----------------------|-------------------|--------------|
| J1  | F1       | O₁                   | M₁,M₂               | Before F₄             | (30,32,34),(29,32,35) | (160 230 290 310) |
|     |          | O₂,O₃               | M₃,M₄               |                       | (16,17,19),(15,18,20) |               |
|     |          |                      | M₅                   |                       | (14,16,17)         |              |
| F₂  | O₄       | M₁,M₂               |                      |                       | (22,24,26),(21,23,27) |               |
|     |          | O₅,O₆               | M₃,M₄               |                       | (10,12,13),(11,13,14) |              |
|     |          |                      | M₅                   |                       | (11,13,15)         |              |
| F₃  | O₇       | M₁,M₂               |                      |                       | (15,17,18),(14,16,18) |               |
|     |          | O₈,O₉               | M₁                   |                       | (12,14,16)         |              |
|     |          |                      | M₂                   |                       | (14,16,18)         |              |
| F₄  | O₁₀      | M₁,M₂               |                      |                       | (26,28,30),(27,28,29) |               |
|     |          | O₁₁                 | M₃,M₄               |                       | (10,12,14),(11,13,15) |               |
| J2  | F₁       | O₁,O₂               | M₁,M₂               | Before F₅             | (8,10,13),(9,12,15) | (170 240 280 300) |
|     |          |                      | M₃,M₄               |                       | (10,11,13),(7,9,12) |              |
|     |          |                      | M₅                   |                       | (11,13,15)         |              |
| F₄  | O₇       | M₁,M₂               |                      |                       | (11,13,16),(14,16,17) |               |
|     |          | O₈,O₉               | M₃,M₄               |                       | (12,15,17),(11,15,16) |              |
|     |          |                      | M₅                   |                       | (20,22,24),(21,23,25) |              |
| F₅  | O₁₀      | M₁,M₂               |                      |                       | (20,23,25)         |              |
|     |          | O₁₁                 | M₃,M₄               |                      | (14,16,18),(9,12,14) |               |
| J₃  | F₁       | O₁                   | M₃,M₄               | Before F₄             | (22,24,26),(21,23,25) | (200 230 280 310) |
|     |          |                      | M₅                   |                       | (14,16,18),(15,17,19) |               |
|     |          |                      |                      |                       | (15,17,18)         |              |
| F₃  | O₄       | M₁,M₂               |                      |                       | (30,32,35),(29,31,34) |               |
|     |          | O₅,O₇               | M₃                   |                      | (20,22,24),(21,23,25) |              |
|     |          |                      |                      |                       | (13,15,17)         |              |
| J₄  | F₁       | O₁                   | M₃,M₄               | Before F₄             | (12,15,17),(13,15,17) | (180 230 270 300) |
|     |          |                      | M₅                   |                       | (30,32,34),(29,31,33) |               |
|     |          |                      |                      |                       | (29,32,34),(30,32,34) |              |
| F₃  | O₃       | M₁,M₂               |                      |                       | (24,26,28),(20,22,24) |               |
|     |          | O₅,O₇               | M₃,M₄               |                       | (11,13,15),(9,11,13) |              |
|     |          |                      | M₅                   |                       | (12,13,15)         |              |
| F₄  | O₇       | M₁,M₂               |                      |                       | (14,16,18),(13,16,18) |               |
|     |          | O₈,O₉               | M₃,M₄               |                       | (7,9,11),(8,10,12)  |              |
|     |          |                      | M₅                   |                       | (18,20,23),(15,17,20) |              |
| F₅  | O₁₀      | M₁,M₂               |                      |                       | (9,12,14)         |              |
|     |          | O₁₁                 | M₃                   |                      | (9,11,13)         |              |
| J₅  | F₁       | O₁                   | M₃,M₄               | Before F₅             | (23,25,26),(22,24,26) | (210 260 290 310) |
|     |          |                      | M₅                   |                       | (15,16,17),(14,16,18) |               |
|     |          |                      |                      |                       | (15,17,18),(13,15,17) |              |
| F₃  | O₄       | M₁,M₂               |                      |                       | (9,12,14)         |              |
|     |          | O₅,O₇               | M₃,M₄               |                       | (23,25,26),(24,26,28) |              |
|     |          |                      | M₅                   |                       | (11,13,15),(10,12,14) |              |
| F₅  | O₈,O₉   | M₃,M₄               |                      |                       | (12,14,16)         |              |
|     |          |                      | M₅                   |                       | (16,17,18),(15,17,19) |              |
|     |          |                      |                      |                       | (16,18,20)         |              |
| F₆  | O₁₀      | M₁,M₂               |                      |                       | (32,34,26),(31,34,35) |               |
|     |          | O₁₁                 | M₃,M₄               |                      | (12,14,16),(11,15,17) |              |
0.5, it shows that the solution set obtained by the MLCO method is closer to the real Pareto front end. Comparing the IGD values, in the 32 groups of experiments, when using the MLCO method, the IGD values of 31 groups of experiments were all less than the obtained IGD values by NSGAII, NSGAII + N5, and the IGD value of 19 groups of experiments were 0. The experimental results indicated that the MLCO method can effectively solve FMOIPPS problems and demonstrate the better performance than NSGAII and NSGAII + N5.

### Calculation results and discussion

1. The number and proportion of Pareto solutions.

The solution sets optimized by the three algorithms are combined, and the global non-dominated solution set is obtained according to the dominant relationship. Calculate the number and percentage of global non-dominated solutions for each algorithm. The results of Pareto solutions obtained by three algorithms are shown in Table 8.

According to the results in Table 8, the number of non-dominated solutions obtained by the MLCO method is more than the number of non-dominated solutions obtained by NSGAII and NSGAII + N5. In addition, as the scale of the problem increases, the number of non-dominated solutions obtained by the MLCO method becomes more and more. In the case of the same number of machines, with the increase of the number of jobs, the proportion of non-dominated solutions obtained by the MLCO method is also increasing.

\[ \text{accuracy} = \frac{1}{m} \sum_{i=1}^{m} \frac{|f_i - \text{excellent}_i|}{|\text{excellent}_i - \text{worsen}_i|} \]  

(7)

In the formula (7), \( \text{accuracy} \) represents the optimization performance of optimization algorithm, \( m \) is the number of optimization objectives, \( f_i \) represents the number of the objectives. \( \text{excellent}_i \) and \( \text{worsen}_i \) are the best values and worst value respectively. The smaller the value of \( \text{accuracy} \), the better the optimization result. The solution sets obtained by the three algorithms are

### Table 5. The parameters in the proposed algorithm.

| Parameters       | Process planning | Process section | Scheduling |
|------------------|------------------|-----------------|------------|
| Population size  | 50               | 50              | 100        |
| Number of generations | 30           | 50              | 100        |
| Probability of mutation | 0.10      | 0.10            | 0.20       |
| Probability of crossover | 0.80        | 0.80            | 0.80       |

### Table 6. Reference point for experiment 1.

| No. | \( f_1 \) | \( f_2 \) | \( f_3 \) | No. | \( f_1 \) | \( f_2 \) | \( f_3 \) |
|-----|----------|----------|----------|-----|----------|----------|----------|
| 1   | 172      | 191      | 211      | 22  | 167      | 183      | 198      |
| 2   | 190      | 205      | 221      | 23  | 172      | 190      | 205      |
| 3   | 196      | 215      | 239      | 24  | 189      | 205      | 219      |
| 4   | 201      | 222      | 249      | 25  | 188      | 205      | 220      |
| 5   | 204      | 227      | 247      | 26  | 191      | 208      | 224      |
| 6   | 183      | 201      | 216      | 27  | 199      | 221      | 240      |
| 7   | 180      | 196      | 212      | 28  | 198      | 219      | 237      |
| 8   | 181      | 197      | 212      | 29  | 200      | 217      | 236      |
| 9   | 167      | 188      | 207      | 30  | 198      | 217      | 236      |
| 10  | 183      | 203      | 223      | 31  | 189      | 213      | 229      |
| 11  | 183      | 202      | 220      | 32  | 185      | 201      | 214      |
| 12  | 176      | 194      | 211      | 33  | 182      | 201      | 219      |
| 13  | 170      | 189      | 210      | 34  | 202      | 227      | 247      |
| 14  | 179      | 198      | 217      | 35  | 175      | 190      | 207      |
| 15  | 183      | 197      | 213      | 36  | 176      | 195      | 212      |
| 16  | 162      | 179      | 195      | 37  | 180      | 195      | 211      |
| 17  | 163      | 180      | 196      | 38  | 184      | 211      | 232      |
| 18  | 164      | 181      | 197      | 39  | 170      | 188      | 206      |
| 19  | 159      | 178      | 193      | 40  | 176      | 195      | 211      |
| 20  | 158      | 175      | 196      | 41  | 184      | 203      | 218      |
| 21  | 164      | 187      | 207      | 42  | 184      | 204      | 218      |

According to the results in Table 8, the number of non-dominated solutions obtained by the MLCO method is more than the number of non-dominated solutions obtained by NSGAII and NSGAII + N5. In addition, as the scale of the problem increases, the number of non-dominated solutions obtained by the MLCO method becomes more and more. In the case of the same number of machines, with the increase of the number of jobs, the proportion of non-dominated solutions obtained by the MLCO method is also increasing.

2. Performance indicator of evaluating optimization results

\[ \text{accuracy} = \frac{1}{m} \sum_{i=1}^{m} \frac{|f_i - \text{excellent}_i|}{|\text{excellent}_i - \text{worsen}_i|} \]  

(7)

In the formula (7), \( \text{accuracy} \) represents the optimization performance of optimization algorithm, \( m \) is the number of optimization objectives, \( f_i \) represents the number of the objectives. \( \text{excellent}_i \) and \( \text{worsen}_i \) are the best values and worst value respectively. The smaller the value of \( \text{accuracy} \), the better the optimization result. The solution sets obtained by the three algorithms are
Table 7. GD and IGD indicators’ average values of three algorithms.

| Problems | NSGAII | NSGAII + N5 | MLCO |
|----------|--------|-------------|------|
|          | GD     | IGD         | GD   | IGD   |
| 5 × 5    | 0.68701 | 0.725167    | 0.523603 | 0.536922 |
| 5 × 10   | 1.22736 | 1.7426      | 1.11677 | 1.33619 |
| 5 × 15   | 1.2357  | 1.52003     | 1.04367 | 1.05681 |
| 5 × 20   | 0.48294 | 0.405397    | 0.495547 | 0.597183 |
| 7 × 7    | 0.312459 | 0.626532   | 0.296996 | 0.335882 |
| 7 × 10   | 0.80271 | 1.07006     | 0.832496 | 1.01293 |
| 7 × 14   | 1.49671 | 1.51279     | 1.44374 | 1.39721 |
| 7 × 21   | 2.17336 | 2.94532     | 2.45204 | 2.38393 |
| 8 × 8    | 0.91947 | 0.899956    | 1.14446 | 0.945515 |
| 8 × 12   | 1.39792 | 1.63744     | 1.37618 | 1.64094 |
| 8 × 16   | 0.95386 | 0.850429    | 0.85708 | 0.824312 |
| 8 × 20   | 0.22387 | 0.253008    | 0.272682 | 0.334704 |
| 10 × 10  | 0.410213 | 0.409253  | 0.320932 | 0.388652 |
| 10 × 15  | 1.34111 | 1.47182     | 1.124  | 1.1203 |
| 10 × 20  | 1.32736 | 1.63413     | 1.06918 | 1.62524 |
| 10 × 25  | 1.17246 | 1.70079     | 1.60488 | 2.47439 |
| 12 × 12  | 0.778431 | 0.953232   | 0.751103 | 1.00417 |
| 12 × 18  | 0.606994 | 0.565294    | 0.706244 | 0.508206 |
| 12 × 24  | 0.938617 | 0.89353     | 0.802856 | 0.966715 |
| 12 × 30  | 0.524848 | 1.21249     | 0.511267 | 1.4394 |
| 15 × 15  | 1.12171 | 1.25053     | 1.23618 | 1.26972 |
| 15 × 23  | 1.26026 | 1.10755     | 0.951435 | 1.07226 |
| 15 × 30  | 1.0849  | 1.17926     | 1.05238 | 1.73432 |
| 15 × 37  | 0.731565 | 0.594055    | 0.55088 | 0.687075 |
| 17 × 17  | 1.09156 | 1.18851     | 1.11793 | 1.27613 |
| 17 × 26  | 1.44233 | 0.944394    | 0.985217 | 1.07193 |
| 17 × 34  | 1.2975  | 1.95782     | 1.01511 | 1.97579 |
| 17 × 42  | 0.634338 | 0.430055   | 0.953792 | 1.45164 |
| 20 × 20  | 1.98488 | 3.76583     | 1.70074 | 2.38991 |
| 20 × 30  | 1.96863 | 2.53111     | 1.37009 | 1.8453 |
| 20 × 40  | 2.24895 | 2.75901     | 2.09016 | 2.36601 |
| 20 × 50  | 1.00664 | 2.25252     | 1.26481 | 2.51362 |

The bold-faced values indicate that the values obtained by MLCO are better than the values obtained by NSGA-II and NSGA-II+N5.

Table 8. The number and proportion of Pareto solutions obtained by three algorithms.

| Problems | Pareto solutions | NSGAII | NSGAII + N5 | MLCO |
|----------|-----------------|--------|-------------|------|
|          | Number | Proportion | Number | Proportion | Number | Proportion |
| 5 × 5    | 43     | 16.3% | 10 | 21.3% | 26 | 60.5% |
| 5 × 10   | 54     | 100% | 0 | 0 | 54 | 100% |
| 5 × 15   | 66     | 100% | 0 | 0 | 66 | 100% |
| 5 × 20   | 184    | 100% | 0 | 0 | 184 | 100% |
| 7 × 7    | 96     | 27 | 28.1% | 27 | 28.1% | 42 | 43.8% |
| 7 × 10   | 69     | 1 | 1.4% | 1 | 1.4% | 68 | 98.6% |
| 7 × 14   | 43     | 100% | 0 | 0 | 43 | 100% |
| 7 × 21   | 41     | 100% | 0 | 0 | 41 | 100% |
| 8 × 8    | 39     | 7.7% | 0 | 0 | 41 | 100% |
| 8 × 12   | 45     | 7.7% | 0 | 0 | 45 | 100% |
| 8 × 16   | 112    | 100% | 0 | 0 | 112 | 100% |
| 8 × 20   | 330    | 12 | 3.63% | 0 | 0 | 318 | 96.3% |
| 10 × 10  | 139    | 3.6% | 5 | 3.6% | 129 | 92.8% |
| 10 × 15  | 59     | 100% | 0 | 0 | 59 | 100% |
| 10 × 20  | 63     | 100% | 0 | 0 | 63 | 100% |
| 12 × 25  | 33     | 100% | 0 | 0 | 33 | 100% |
| 12 × 25  | 26     | 100% | 0 | 0 | 26 | 100% |
| 12 × 25  | 89     | 100% | 0 | 0 | 89 | 100% |
| 12 × 24  | 64     | 100% | 0 | 0 | 64 | 100% |
| 12 × 30  | 117    | 100% | 0 | 0 | 117 | 100% |

(continued)
combined, and the optimal value of each optimization objective is found from the combined solution set. The solution set is obtained from each algorithm, and the solution with the best comprehensive performance is selected as the optimal solution of the algorithm. The results are obtained according to the calculation formula of accuracy, as shown in Table 9.

As shown in the results of Table 9, the accuracy values of 32 sets problem examples are obtained by using the MLCO. It found that the accuracy values of 30 experiments were less than the accuracy values obtained by NSGAII and NSGAII + N5, and the accuracy values of 14 experiments were less than 0.1. It showed that the algorithm has a high performance of optimization. Table 10 is scheduling scheme for the problem 10×3.

The results of 32 sets of experiments show that the ability of the MLCO method to find the optimal solution is better than the other two methods. According to the data in Tables 7 and 8, it can be seen that the MLCO method has better optimization ability. The MLCO method is divided into three layers to solve the problem step by step. The process planning layer formulates near-optimal process plans for each job. The second layer considers all the processing tasks, selects the process plan for each job, and the third layer arranges processing time for all jobs. Compared with NSGAII and NSGAII + N5, MLCO method can make full use of the data information and processing information of the optimization process, and it also reduces the randomness of the optimization process. Especially with the expansion of the problem scale, reducing the randomness of the optimization is helpful to improve the ability to find the optimal solution. In addition, the MLCO method adopts a boundary search strategy in the third layer to enhance the search ability of boundary solutions. The results in Table 9 show that the accuracy value of the MLCO method is smaller, indicating that the quality of the non-dominated solutions obtained by the MLCO method is better. The individual comprehensive evaluation strategy is adopted in MLCO method to select the non-dominated solutions and improve the quality of non-dominated solutions. The experimental results show that the MLCO method proposed in this paper is effective for solving the FMOIPPS problem.

### Conclusion and future works

In view of the complexity of actual production companying many uncertain factors, the diversity of demands and uncertain factors are comprehensively considered in this paper. For the multi-objective IPPS problem under uncertain conditions, the triangular fuzzy number is used to describe the uncertain processing time, and the trapezoid fuzzy number is adopted to indicate the uncertain due date. Minimize fuzzy makespan, maximize average customer satisfaction and minimize the spread of fuzzy makespan are planned as optimization objective. MLCO method is proposed to solve the FMOIPPS problem. The algorithm uses three layers of optimization methods including process planning layer, process selection layer, and scheduling layer. A sharing function strategy was adopted to maintain the population. A boundary search strategy is proposed to improve the search ability of the algorithm. An individual comprehensive evaluation strategy is proposed to selecting non-dominated solutions. The feasibility and effectiveness of the MLCO method are proved by 32 sets of examples with different scales. The results of the examples show that the algorithm has good practicability in solving uncertain multi-objective IPPS problems.

However, the actual workshop production process is often accompanied by many uncertainties, and more uncertainties should be considered in the IPPS problem. To solve the more complex and uncertain multi-objective IPPS problem and how to study the dynamic IPPS problem become the direction of further research in this paper.
The bold-faced values indicate that the values obtained by MLCO are better than the values obtained by NSGA-II and NSGA-II+N5.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research work is supported by the National Natural Science Foundation of China (grant no. 51905494 and 51775517), MOE (Ministry of Education in China) Project of Humanities and Social Sciences (grant no. 19YJCZH185), Key Scientific and Technological Research Project in Henan Province (grant no. 202102210088).

ORCID iD
Xiaoyu Wen https://orcid.org/0000-0002-0716-1101

References
1. Sugimura N, Hino R and Moriwaki T. Integrated process planning and scheduling in holonic manufacturing systems. In: Proceedings of the 2001 IEEE international symposium on assembly and task planning, Fukuoka, Japan, 29 May 2001, pp. 250–255. Piscataway, NJ: IEEE.
2. Haddadzade M, Razifar MR, Zarandi MH, et al. Integration of process planning and job shop scheduling with stochastic processing time. Int J Adv Manuf Technol 2014; 71(1): 241–252.
3. Phanden RK, Jain A, Verma R, et al. An approach for integration of process planning and scheduling. Int J Comput Integr M 2013; 26(4): 284–302.
4. Li XY, Gao L, Zhang CY, et al. A review on integrated process planning and scheduling. Int J Manuf Res 2010; 5(2): 161–180.
5. Chryssolouris G, Chan S, Cobb W, et al. Decision making on the factory floor: an integrated approach to process planning and scheduling. Robot Cim-Int Manuf 1984; 315–319.
6. Phanden RK, Jain A, Verma R, et al. Integration of process planning and scheduling: a state-of-the-art review. Int J Comput Integr M 2011; 24(6): 517–534.
7. Zhang S and Wong TN. Integrated process planning and scheduling: an enhanced ant colony optimization heuristic with parameter tuning. J Intell Manuf 2018; 29(3): 585–601.
8. Li XY, Gao L, Pan QK, et al. An effective hybrid genetic algorithm and variable neighborhood search for integrated process planning and scheduling in a packaging machine workshop. IEEE Trans Syst Man Cybernetics Syst 2019; 49(10): 1933–1944.
9. Shao XY, Li XY, Gao L, et al. Integration of process planning and scheduling: a modified genetic algorithm-based approach. Comput Oper Res 2009; 36(6): 2082–2096.
10. Guo YW, Li WD, Mileham AR, et al. Applications of particle swarm optimization in integrated process

Table 10. The scheduling scheme of the problem $10 \times 5$

| Jobs | Operations | Machines | Start time | End time | Jobs | Operations | Machines | Start time | End time |
|------|------------|----------|------------|----------|------|------------|----------|------------|----------|
| J1   | O1         | M2       | 0          | 29 32 35 | O3   | M4         | 218 239 258 | 231 254 275 |
| O4   | 29 32 35   | 50 55 62 |           |          | O6   | M2         | 59 67 76   | 77 87 99   |
| O10  | 50 55 62   | 77 83 91 |           |          | O3   | M2         | 77 87 99   | 98 111 126 |
| O7   | 123 132 139| 138 149 157|          |          | O2   | M1         | 200 216 231| 209 227 244 |
| O12  | 138 149 157| 148 161 171|          |          | O7   | M2         | 0          | 15 17 18   |
| O12  | 184 200 214| 204 222 239|          |          | J6   | M2         | 42 47 52   | 65 71 78   |
| J2   | O1         | M1       | 0          | 8 10 13  | O13  | M3         | 65 71 78   | 80 89 98   |
| O2   | 9 11 13    | 19 22 26 |           |          | O12  | M5         | 80 89 98   | 100 111 122 |
| O3   | 80 89 98   | 92 104 115|          |          | O9   | M1         | 112 120 126| 123 132 139|
| O12  | 113 128 146| 122 140 160|          |          | O10  | M2         | 123 132 139| 133 144 154|
| O6   | 158 170 181| 169 183 196|          |          | O15  | M2         | 144 159 172|           |
| O7   | 169 183 196| 181 198 212|          |          | O1   | M4         | 217 238 256| 229 252 272|
| O3   | 181 198 212| 197 216 232|          |          | O8   | M1         | 50 55 62   | 59 67 76   |
| O11  | 197 216 232| 217 239 257|          |          | O2   | M3         | 59 67 79   | 74 83 93   |
| J3   | O1         | M4       | 29 31 33   | 50 54 58 | O3   | M3         | 92 104 115| 113 127 140|
| O3   | 113 127 140| 125 142 157|          |          | O5   | M1         | 184 198 211| 200 216 231|
| O5   | 153 174 195| 174 197 220|          |          | O6   | M2         | 204 223 239| 217 238 256|
| O2   | 227 247 265| 256 278 299|          |          | O9   | M3         | 217 239 257| 231 255 275|
| J4   | O1         | M4       | 0          | 29 31 33 | O11  | M5         | 231 255 275| 251 277 299|
| O11  | 29 31 33   | 59 63 69 |           |          | O6   | M3         | 30 34 37   | 44 50 55   |
| O8   | 98 111 126 | 113 128 146|          |          | O8   | M3         | 44 50 55   | 54 62 70   |
| O10  | 169 183 196| 184 200 214|          |          | O4   | M2         | 137 147 156| 168 180 191|
| O7   | 184 200 214| 204 222 238|          |          | O2   | M3         | 204 222 238| 227 247 265|
| O11  | 229 252 272| 236 261 283|          |          | J10  | M2         | 42 47 52   | 65 71 78   |
| O6   | 236 261 283| 250 277 301|          |          | O13  | M3         | 15 17 18   | 30 34 37   |
| J5   | O1         | M3       | 19 22 26   | 42 47 52 | O3   | M3         | 82 88 92   | 112 120 126|
| O3   | 59 63 66   | 82 88 92 |           |          | O9   | M4         | 112 120 126| 137 147 156|
| O11  | 82 88 92   | 93 103 109|          |          | O2   | M4         | 137 147 156| 158 170 181|
| O10  | 122 140 160| 153 174 195|          |          | O8   | M1         | 158 170 181| 184 198 211|
| O4   | 209 227 244| 218 239 258|          |          | O11  | M5         | 184 198 212| 196 213 229|

1900 Measurement and Control 53(9-10)
planning and scheduling. *Robot Comput Integr Manuf* 2009; 25(2): 280–288.
11. Li XY, Shao XY, Gao L, et al. An effective hybrid algorithm for integrated process planning and scheduling. *Int J Prod Econ* 2010; 126(2): 289–298.
12. Li XY, Gao L, Shao XY, et al. Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling. *Comput Oper Res* 2010; 37(4): 656–667.
13. Wen L, Li XY, Gao L, et al. A new two-Level hierarchical diagnosis network based on convolutional neural network. *IEEE T Instrum Meas* 2020; 69(2): 330–338.
14. Wen L, Li XY, Gao L, et al. A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Comput Appl* 2019; 1–14.
15. Zhang S, Yu ZN, Zhang WY, et al. An extended genetic algorithm for distributed integration of fuzzy process planning and scheduling. *Math Probl Eng* 2016; 1–13.
16. Li XY, Gao L, Wang WW, et al. Particle swarm optimization hybridized with genetic algorithm for uncertain integrated process planning and scheduling with interval processing time. *Comput Ind Eng* 2019; 135: 1036–1046.
17. Wen XY, Li XY, et al. Modified honey bees mating optimization algorithm for multi-objective uncertain integrated process planning and scheduling problem. *Int J Adv Robot Syst* 2020; 1–17.
18. Liu BJ, Fan YS, Liu Y, et al. A fast estimation of distribution algorithm for dynamic fuzzy flexible job-shop scheduling problem. *Comput Ind Eng* 2015; 87: 193–201.
19. Palacios JJ, González-Rodríguez I, Vela CR, et al. Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop. *Fuzzy Set Syst* 2015; 278: 81–97.
20. Lei D. Fuzzy job shop scheduling problem with availability constraints. *Comput Ind Eng* 2010; 58(4): 610–617.
21. Jain A, Jain P, Singh IP, et al. An integrated scheme for process planning and scheduling in FMS. *Int J Adv Manuf Tech* 2006; 30(11): 1111–1118.
22. Li XY, Zhang CY, Gao L, et al. An agent-based approach for integrated process planning and scheduling. *Expert Syst Appl* 2010; 37(2): 1256–1264.
23. Chu YF, You FQ, Wassick JM, et al. Integrated planning and scheduling under production uncertainties: bi-level model formulation and hybrid solution method. *Comput Chem Eng* 2015; (72): 255–272.
24. Sekerci A, Erol S, Botsali R, et al. A neuro-fuzzy model for a new hybrid integrated process planning and scheduling system. *Expert Syst Appl* 2013; 40(13): 5341–5351.
25. Li XY, Gao L, Shao XY, et al. An active learning genetic algorithm for integrated process planning and scheduling. *Expert Syst Appl* 2012; 39(8): 6683–6691.
26. Leung CW, Wong TN, Mak KL, et al. Integrated process planning and scheduling by an agent-based ant colony optimization. *Comput Ind Eng* 2010; 59(1): 166–80.
27. Zhang L and Wong TN. An object-coding genetic algorithm for integrated process planning and scheduling. *Eur J Oper Res* 2015; 244(2): 434–444.
28. Sobeyko O and Moench L. Integrated process planning and scheduling for large-scale flexible job shops using metaheuristics. *Int J Prod Res* 2017; 55(2): 392–409.
29. Shokouhi E. Integrated multi-objective process planning and flexible job shop scheduling considering precedence constraints. *Prod Manuf Res* 2018; 6(1): 61–89.
30. Li XY, Gao L, Li WD, et al. Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling. *Expert Syst Appl* 2012; 39(1): 288–297.
31. Luo GF, Wen XY, Li H, et al. An effective multi-objective genetic algorithm based on immune principle and external archive for multi-objective integrated process planning and scheduling. *Int J Adv Manuf Tech* 2017; 91(9): 3145–3158.
32. Li XY, Lu C, Gao L, et al. An effective multi-objective algorithm for energy efficient scheduling in a real-life welding shop. *IEEE Trans Indus Inform* 2018; 14(12): 5400–5409.
33. Yin LJ, Li XY, Gao L, et al. A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem, considering productivity, energy efficiency and noise reduction. *Sustain Comput Inform* 2017; 13: 15–30.
34. Lu C, Gao L, Li XY, et al. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Eng Appl Artif Intel* 2017; 57: 61–79.
35. Lu C, Gao L, Li XY, et al. Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *J Clean Prod* 2017; 144(144): 228–238.
36. Gao KZ, Suganthan P N, Pan QK, et al. An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time. *Expert Syst Appl* 2016; (65): 52–67.
37. Jamrus C, Chien MG and Sethanan K. Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing. *IEEE T Sustain Comput* 2018; 3(1): 32–41.
38. Lei D. Population-based neighborhood search for job shop scheduling with interval processing time. *Comput Ind Eng* 2011; 61(4): 1200–1208.
39. Lei D. Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling. *Appl Soft Comput* 2012; 12(8): 2237–2245.
40. Hu YM, Yin MH, Li XT, et al. A novel objective function for job-shop scheduling problem with fuzzy processing time and fuzzy due date using differential evolution algorithm. *Int J Adv Manuf Tech* 2011; 56(9): 1125–1138.
41. Gao KZ, Suganthan P N, Pan QK, et al. An effective discrete harmony search algorithm for flexible job shop scheduling problem with fuzzy processing time. *Int J Prod Res* 2015; 53(19): 5896–5911.
42. Gao KZ, Suganthan PN, Pan QK, et al. Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion. *Knowl-Based Syst* 2016; 109(109): 1–16.
43. Lu C, Li XY, Gao L, et al. An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times. *Comput Ind Eng* 2017; 104: 156–174.
44. Li XY, Gao L, Wen XY, et al. Application of an efficient modified particle swarm optimization algorithm for process planning. *Int J Adv Manuf Tech* 2013; 67: 1355–1369.
45. Schoenauer M and Michalewicz Z. Boundary operators for constrained parameter optimization problems. In: *Proceedings of the 7th international conference on genetic algorithms*, MI, USA, July 19–23, 1997, pp.322–329. Morgan Kaufmann.