Fuzzy Target Job Shop Scheduling Based on Improved Multi-population Genetic Algorithm with Clustering

Yi Bu, Hui-Zhi Yu, Rui-Kai Zhu

School of Computer, Nanjing University of Information Science & Technology, Nanjing 210044, China

Abstract. Owing to the traditional genetic algorithm is difficult to hit the optimal solution in the optimization problem of fuzzy target shop scheduling, this paper converts the weighting of the two goals of processing completion time and customer satisfaction into a single objective task, and adds fuzzy completion time constraints on the basis of traditional shop scheduling constraints A fuzzy objective shop scheduling model is constructed, some genetic operators using different strategies are designed and a multi-population genetic algorithm based on clustering is proposed according to the similarity of populations The algorithm first generates multiple initial populations by extending the integer encoding of the processing order and processing machines, and each population undergoes independent genetic operations and then forms importation, integration, and artistic selection to share the excellent genes of individuals The algorithm is verified through experimental simulations that it has strong stability and global search capability: The probability of finding the global optimal solution has increased to 61%.

Keywords: genetic algorithm; K-means cluster; job shop scheduling; fuzzy target.

1. Introduction

Workshop scheduling is one of the important links in the production process. It involves how to arrange production tasks and resources to maximize production efficiency and reduce costs. However, in practical scheduling problems, the processing time of the workpiece needs to be within the time window requested by the customer, so the processing completion time often follows the fuzzy submission time required by the customer's delivery time. In addition, for simple optimization problems, chromosomes can fully express the potential solutions of the problem, but due to the uncertainty and complexity of the production environment, the complexity of the problem has significantly increased, and single-layer chromosome coding can no longer fully express the content of the solution. Therefore, workshop scheduling problems often present difficulties such as dynamic fuzziness and multi-constraints, and are optimization problems with fuzzy objectives.

To solve this problem, in recent years, more and more researchers have begun to explore emerging optimization methods to improve traditional genetic algorithms. Different researchers have provided a combination of multiple methods for coding methods and genetic operators. Among them, the improved multi-population genetic algorithm is a meta-heuristic algorithm that improves the coding method and population generation strategy of traditional genetic algorithms and introduces immigration operators, manual selection operators, merging operators, and elite strategies. Through multi-layer coding, the individual code is divided into multiple layers, with each layer representing different meanings, and the solution to the problem is expressed through the code values of all layers. To improve search efficiency and the interpretability of optimization results, the application field of genetic algorithms has been expanded, making it convenient for solving complex problems. After adding different genetic operators and adjusting the communication period between populations, the global search ability of the problem improved, but the search efficiency decreased.

The first section of this article introduces the relevant methods and problem descriptions of fuzzy objective workshop scheduling problems. Section 2 introduces the theoretical knowledge required for this article, including multi-layer coding, fuzzy mathematics, genetic algorithms, and their different genetic operators. In Section 3, this article constructs the basic process of an improved multi-population genetic algorithm and verifies through comparative experiments that this method has significant advantages in improving workshop production efficiency, reducing costs, and adapting to uncertainty. Finally, summarize the entire article and propose prospects.

2. Workshop scheduling

Most existing workshop scheduling methods are based on rule-based methods or meta-heuristic algorithms, and the model structure is too single to solve more complex problems. The commonly used rule-based methods include: first in first out (FIFO), first in last out (LIFO), shortest processing time (SPT), shortest total processing time (STPT), minimum remaining operations (LOR), and next task minimum waiting operation (LQNO). These are methods based on simple rules, which prioritize processing workpieces, processing equipment, and transportation equipment according to certain criteria to provide a suitable solution. These methods are simple and easy to use, but usually cannot find the global optimal solution. However, for workshop scheduling with fuzzy objectives, its dynamic fuzziness and multiple constraints make traditional rule-based optimization algorithms have significant limitations and are difficult to fully utilize. In contrast, meta-heuristic algorithms are a kind of more complex optimization algorithm that searches for local optimal solutions on workshop scheduling problems through different optimization iteration operators. This algorithm can achieve higher accuracy and better global search ability than rule-based methods but also has higher computational complexity.

Kiran et al. divided the objectives of job shop scheduling into three categories: delivery time, completion time, and cost and provided multiple scheduling objectives. The delivery date is generally determined according to the wishes of customers. However, in actual production, due to some external factors, such as natural environment, socio-economic and personal reasons, the problem model cannot be built on the basis of accurate parameters. However, there are also great limitations in simulating problems with randomly varying parameters. Therefore, fuzzy sets and fuzzy logic are increasingly introduced into workshop scheduling problems to deal with uncertain goals.

2.1 Research on Workshop Scheduling

In recent years, scholars have been continuously researching workshop scheduling problems. Yuan Kun et al. solved the fuzzy objective flexible workshop scheduling problem based on traditional genetic algorithms[1]; Liu Aijun and others studied multi-objective fuzzy flexible workshop scheduling using multiple population genetic algorithms [2]; Yang Fan et al. applied a new data structure to improve the genetic algorithm and studied its application in workshop scheduling problems[3]; Zhong Huichao introduced clustering algorithm and combined Reinforcement learning with genetic algorithm to study workshop scheduling method[4]; Chanas et al. studied the minimum "maximum delay time" job shop scheduling problem for a single machine tool[5] and the minimum "fuzzy delivery delay time" problem for two machines[6]; Itoh et al. studied the "minimum number of delayed jobs" scheduling problem with fuzzy set "completion date" and "delivery date"[7]; Masatoshi et al. studied the job shop scheduling problem of "fuzzy processing time" and "fuzzy delivery time"[8].

According to the above research, the main contribution of this paper lies in the combination of an unsupervised learning clustering algorithm and multi-population genetic algorithm, the design of an efficient multi-level coding method and population selection strategy, and the introduction of customer fuzzy satisfaction to measure the latest delivery date to solve fuzzy multi-objective job shop scheduling problems. This method not only has advantages in search efficiency and interpretability of optimization results but also achieves good results in practical applications. Therefore, this article believes this method has broad application prospects and can provide strong support and assistance for optimizing production scheduling.

2.2 Problem Description of Basic Workshop Scheduling

The traditional workshop scheduling problem can be described mathematically as having n parts that need to be processed on m machines, and a time matrix T, $t_{ij} \epsilon T$ is set to represent the time when the i-th part p_i uses the j-th machine to process the part. Generally, the following model can be defined:

$$\min f = \max_{1 \le i \le n} C_i \,, \tag{1}$$

the above equation is the objective function with the minimum completion time as the goal, in which f is the minimum performance indicator of the maximum completion time. Among them, the time matrix of the machine processing part is $C, c_{ij} \epsilon C$, representing the completion time of the i-th part p_i on the j-th machine.

Let the machine set and parts set as follows:

$$M = \{m_1, m_2, \dots, m_m\}, \tag{2}$$

$$P = \{p_1, p_2, \dots, p_n\}, \tag{3}$$

where m_j represents the j-th machine and p_i represents the i-th part, $j = 1,2,3, \dots m$ $i = 1,2,3, \dots n$.

Let the process sequence set as:

$$OP = \begin{pmatrix} op_{i1} & \cdots & op_{ik} \\ \vdots & \ddots & \vdots \\ op_{n1} & \cdots & op_{nk} \end{pmatrix}, \tag{4}$$

where $\{op_{i1}, op_{i2}, \dots, op_{ik}\}$ represents the process sequence of the part p_i and $|op_i|$ represents the number of required processes.

Let the optional machine set as:

$$OPM = \{op_{i1}, op_{i2}, \dots, op_{ik}\}, \tag{5}$$

where $op_{ij} = \{op_{ij1}, op_{ij2}, \dots, op_{ijk}\}$ represents the processing machine that can be selected for the process j of part p_i.

Generally speaking, workshop scheduling satisfies the following basic assumptions:

(I) The number of times each machine is used for each process of each component shall not exceed 1;

(II) Each part is processed in a certain order;

(III) A certain process can only be processed by one machine at a certain time;

(IV) Each machine can only process one workpiece at a time;

(V) The machining process of a workpiece on the machine cannot be interrupted, that is, one machining task must be completed before the next workpiece can be accepted.

The above model should meet the following constraints:

$$\sum_{k=1}^{op_{ih}} x_{ihk} = 1, i = 1,2,3, \dots, n, \tag{6}$$

Where $x_{ihk} = \begin{cases} 1, & \text{the step h of the workpiece i can be processed on the k machine k} \\ 0, & \text{other} \end{cases}$, the machine uniqueness constraint under constraint (6) indicates that a machine can only handle one process at a time, which is the mathematical expression of hypothesis (III).

$$C_{ij} - S_{ij} - x_{ihk} t_{ij} = 0, \forall i = 1,2, \dots, n, \forall j = 1,2, \dots, m, \tag{7}$$

Where the matrix S represents the start processing time of the i-th part p_i on the j-th machine, and constraint (7) indicates that, under the condition of allowing processing, the completion time of the workpiece processing minus the start processing time is equal to the workpiece processing time. This is the mathematical expression of hypothesis (V).

$$S_{ij} \ge 0 \ \forall i \in \{1,2,3, \dots n\}, \forall j \in \{1,2,3, \dots m\}, \tag{8}$$

$$S_{iJ+1} - C_{ij} \ge 0 \ \forall i \in \{1,2,3, \dots n\}, \forall j \in \{1,2,3, \dots j_i - 1\}, \tag{9}$$

Constraint (8) and (9) are basic constraints that indicate that the starting processing time of each workpiece is greater than or equal to 0, and the starting processing time of the next workpiece on a certain machine must be after the completion time of the previous workpiece. This is the mathematical expression of hypothesis (IV).

$$\begin{cases} C_{i'j'} - C_{ij} + H(Y_{iji'j'}) \geq t_{ij} \\ C_{i'j'} - C_{ij} + H(1 - Y_{iji'j'}) \geq t_{ij} \\ \forall (i,k).(i',k')\ op_{ik} \in OP, op_{i'k'} \in OP \end{cases}, \quad (10)$$

Where $Y_{iji'j'} = \begin{cases} 1, if\ op_{ik} \text{ is prior to } op_{i'k'} \\ 0, \text{other} \end{cases}$, its different values suppress different aspects of constraint (10).

3. Genetic Algorithm Based on Multi Group & Multilayer Coding

The genetic algorithm with multi-group and multi-layer coding[13] is an improvement of the classical genetic algorithm. Although the implementation details vary, it all has the following structure: the algorithm iteratively updates one or more hypothesis pools, i.e. the population Each iteration of evolution evaluates all members of the population based on fitness and then performs genetic operations using a certain strategy to obtain the optimal solution of the problem.

3.1 Coding

The solution of workshop scheduling is a relatively complex structure that includes machine processing orders and machine usage orders. Therefore, chromosomes cannot only encode a single aspect of feasible solutions, and it is necessary to maintain both machine processing order and machine usage order simultaneously. And multi-layer encoding is essentially an extended integer encoding: when the total number n of workpieces to be processed and the number of processing steps of the workpieces n_i is m_j, each chromosome is represented as length:

$$L = 2\sum n_i m_j, \quad (11)$$

An integer string α that can be represented as:

$$\alpha = \begin{bmatrix} \alpha_1 & \cdots & \alpha_i \\ \beta_1 & \cdots & \beta_i \end{bmatrix}, \quad (12)$$

The upper half of the integer string represents the machining sequence of all workpieces on the machine, while the lower half represents the machining machine number of each process of the workpiece.

3.2 Genetic Manipulation

The genetic operations of traditional genetic algorithms include selection, crossover, and mutation. For selection operations, classic algorithms include roulette wheel selection and tournament selection. This article introduces roulette wheel selection with the elite strategy to select chromosomes with good fitness. The probability of individual selection is:

$$pi(i) = \frac{\frac{1}{fitness(i)}}{\sum_{i-1}^{n} \frac{1}{fitness(i)}}, \quad (13)$$

where $pi(i)$ represents the probability of chromosomes i being selected. Roulette wheel selection is essentially a statistical model. Chromosomes with high fitness are more likely to be copied to the next generation than inferior chromosomes, so the solution space will gradually shrink toward the optimal solution. But if the selection probability is set improperly, it may lead to individuals containing high-quality genes being eliminated during the evolution process.

The elite retention strategy is that after each generation of evolution, the elite individuals with the highest fitness in each population are retained by the next generation so that the excellent genes of elite chromosomes can not be randomly selected or replaced by mutation operations in the evolution process, which speeds up the Rate of convergence of the algorithm and the stability of the optimal solution.

The population obtains new chromosomes through the crossing of parental chromosomes, driving the evolutionary development of the entire population. This article adopts the commonly used integer encoding method of integer crossover: randomly selecting 2 parent individuals in each population and generating crossover gene location k. To the first $\sum_{i=1}^{k} n_i m_j$ positions of chromosomes, the processing sequence of certain workpieces represented by the first layer of chromosomes on the machine is crossed. After the completion of chromosome crossing in one layer, the correctness of the first layer of chromosomes is destroyed, and the genes of the second layer of chromosomes need to be recombined and restored. If the processes of certain workpieces are redundant or missing after crossing, the chromosome length needs to be adjusted:

$$\left(\sum_{i=1}^{k} n_i m_j + 1\right) \to 2\sum_{i=1}^{k} n_i m_j \,, \tag{14}$$

To increase the diversity of the population, genetic algorithms obtain new chromosomes through mutation operations. This article adopts two-point mutation to drive population evolution, randomly generating mutation loci and swapping their two levels of gene loci pos_1, pos_2, namely processing steps and corresponding processing machine numbers.

3.3 Immigration Operator and Manual Selection Operator

The multi-population genetic algorithm can search for the optimal solution faster than the traditional genetic algorithm, and the hit of each optimal solution is more stable. Specifically, during the evolutionary process, each population selects, crosses, and mutates independently, and then communicates with each other. If the average solution value of the current population is inferior to the target population, the optimal individual of the current population is set to be attracted to the target population, thus spreading their high-quality genes to the target population, and the inferior individuals in the population will be eliminated as a result. On the contrary, then these two populations randomly exchange individuals with each other. In this process, various populations fuse and communicate with each other, jointly driving individual evolution to achieve optimal goals.

The manual selection operator, on the other hand, manually selects high-quality individuals from each population after immigration operations, controlling population fusion to proceed in the best direction. Specifically, after each population completes a generation of evolution and communication, if the obtained solution is superior to the results of the previous generation, the entire population maintains its current high-quality results. Unlike traditional genetic algorithms, this article does not have the concept of the maximum number of iterations but instead introduces the concept of maximum preserving algebra, which means that only when each population obtains a better solution or maintains the current results, otherwise it will fall back, allowing population fusion to continue. The following figure shows population genetic operations on simulated organisms:

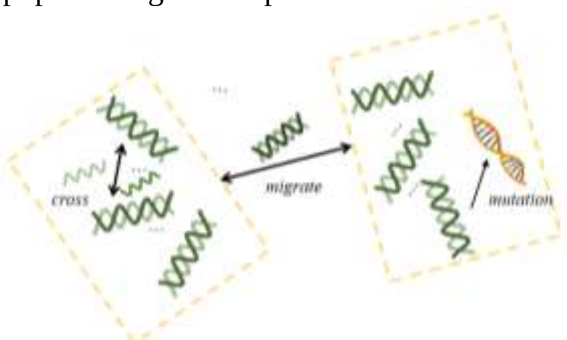

Figure 1 Population diagram

3.4 Merge Operator: K-means clustering[8]

Based on population similarity, when the merging period is reached, K-means clustering is performed to calculate the position of each cluster center and the cosine distance between each cluster center, and to define the diversity value of the population based on this:

$$diversity = \sum \frac{a \cdot b}{||a|| \cdot ||b||}, \tag{15}$$

The basic process of K-means clustering is

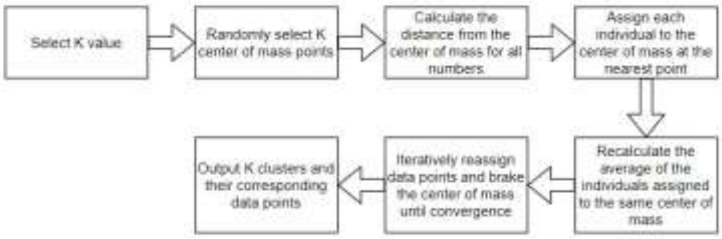

Figure 2 K-means clustering process

When the merging period is reached, the population with the smallest diversity value is merged with the largest population, and the same individuals in the merged large population are deleted to reduce the space and time cost of the algorithm. The following figure shows the population merging operation on a simulated organism:

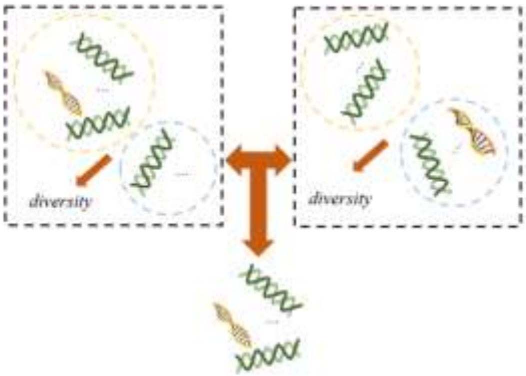

Figure 3 Population Merge

3.5 Customer Fuzzy Satisfaction

In actual workshop scheduling, when the workpiece is processed and delivered often needs to be within the time window specified by the customer. Therefore, for traditional precise delivery times, fuzzy target workshop scheduling improves the completion time of workpieces, specifically by following the delivery fuzzy time required by customers Early delivery or over-delivery can both affect customer satisfaction. Therefore, this article defines the fuzzy delivery date of the workpiece.

The fuzzy distribution function[6] is

$$\mu_d(x) = \begin{cases} 0, x \le d_i^1, x \ge d_i^4 \\ \dfrac{x - d_i^1}{d_i^2 - d_i^1}, d_i^1 < x < d_i^2 \\ \dfrac{d_i^2 - x}{d_i^4 - d_i^3}, d_i^2 < x < d_i^4 \\ 1, d_i^2 < x < d_i^3 \end{cases}, \tag{16}$$

Where $\mu_d(x)$ represents the membership degree of x to the Fuzzy set D_i, and also represents the customer's satisfaction with the delivery date, satisfying $0 \le \mu_d(x) \le 1$. For the submission time of each workpiece, the fuzzy membership of the submission time is calculated as the submission time of the fitness of the next genetic algorithm.

After introducing the concept of fuzzy membership, namely customer satisfaction, traditional workshop scheduling has shifted from a single objective to a multi-objective optimization task. For machine processing, minimize processing time; For customers, maximize satisfaction:

$$\begin{cases}\min f = \max\limits_{1\le i\le n} C_i \\ \max f = \max \sum \mu_d(x)\end{cases}, \tag{17}$$

These are two conflicting objectives that can be weighted and converted into single objective tasks to simplify optimization complexity:

$$Object = \min \beta C_i + \frac{\alpha}{\sum \mu_d(x)}, \tag{18}$$

4. Constructing an Improved Multi-population Genetic Algorithm Framework

The input of the algorithm includes the fitness function used to sort candidate individuals; Maintain population size; The number of populations maintained; the Maximum number of iterations; The hyperparameters that control how a population evolves to produce offspring, i.e. the selection and mutation probabilities of each population in each generation of evolution. The following table provides the algorithm prototype of a genetic algorithm with multiple population multi-layer encoding:

Table 1 Algorithm Prototype of Improved Multi-population Genetic Algorithm

MLC-MPGA (*Fitnss, ps, p n, r, m, max_gen*)

Fitness: Fitness function used to calculate the optimal process and shortest time

p_s: Number of individuals in the population

p_n: Number of population

r: Proportion matrix for cross-selection of offspring in each generation

m: Variability matrix

Max_ Gen: Maximum number of iterations

- Initialize population: randomly generate p_n chromosomes from p_s populations→ P_i
- Evaluation: Calculate the fitness of each individual h in the current population cluster p→ $Fitness(p, h)$
- When the current optimal solution maintains algebra<max_ Gen, do

  Traversing various populations

  Generate offspring Ps of the current population:

  1、Selection operation: Sort the population according to the size of fitness, and use the roulette wheel to select $(1-r)p\_n$ members of P to join Ps
  2、Cross operation: Select $Pr * \frac{p}{2}$ pairs of target individuals from P according to the probability. For each pair of individuals$< h_1, h_2 >$, apply the crossover operator to generate new individuals and add them to the Ps
  3、Mutation operation; The $m\%$ members are selected from Ps by probability. Apply the mutation operator to generate new individuals
  4、Update current population: $Ps \rightarrow P$
  5、Calculate the fitness of each individual in the current population $Fitness(h)$

  When the immigration cycle is reached, do immigration

  When the average fitness value of the target is higher than that of the current population:

  Replace the optimal solution of the current population with the worst solution of the target

  Otherwise:

  Randomly migrate between two populations

Manual selection: Select the optimal solution among all populations

When the consolidation cycle is reached, do

Perform K-means clustering on all populations to calculate their diversity values

Merge population

- Returns the individual with the highest fitness and the objective function value

In practical application, when the optimal value remains unchanged for 5 generations, the algorithm iteration can be stopped. Early stopping to reduce the running time.

4.1 Experimental Data

We conducted experiments on a given dataset, including a table of optional machines for workpieces and a processing schedule for workpieces. Tables 2 and 3 provide detailed information corresponding to the dataset.

Table 2 Experimental Dataset (Process Optional Machine Set)

workpiece	Workpiece 1	Workpiece 2	Workpiece 3	Workpiece 4	Workpiece 5	Workpiece 6
Process 1	3,10	2	3,9	4	5	2
Process 2	1	3	4,7	1,9	2,7	4,7
Process 3	2	5,8	6,8	3,7	3,10	6,9
Process 4	4,7	6,7	1	2,8	6,9	1
Process 5	6,8	1	2,10	5	1	5,8
Process 6	5	4,10	5	6	4,8	3

Table 3 Experimental Dataset (Process Processing Time)

workpiece	Workpiece 1	Workpiece 2	Workpiece 3	Workpiece 4	Workpiece 5	Workpiece 6
Process 1	3,5	6	1,4	7	6	2
Process 2	10	8	5,7	4,3	10,12	4,7
Process 3	9	1,4	5,6	4,6	7,9	6,9
Process 4	5,4	5,6	5	3,5	8,8	1
Process 5	3,3	3	9,11	1	5	5,8
Process 6	10	3,3	1	3	4	3

Table 2 shows the machine numbers that can be selected for each process of each workpiece, while Table 3 corresponds to the time required for each process of each workpiece to be processed on the current machine. According to the completion time of parts processing, customer satisfaction varies. Table 4 provides a distribution table of customer satisfaction time, indicating that customer satisfaction varies during different periods of submission:

Table 4 Experimental Dataset (Time Distribution Table of Customer Satisfaction)

workpiece	d^1	d^2	d^3	d^4
Workpiece 1	40	55	60	70

Workpiece 2	30	40	50	55
Workpiece 3	65	80	90	110
Workpiece 4	45	50	65	75
Workpiece 5	60	70	80	95
Workpiece 6	50	55	70	80

4.2 Experimental Methods

Using the given dataset for experimental simulation, calculate the minimum time required for workpiece processing and the maximum customer satisfaction. Considering different processing situations and customer requirements, the weight setting of these two goals may also vary. Generally speaking, machine processing should be customer-oriented, so in terms of weight setting, the β should be greater than α.

In this experiment, the algorithm is based on Matlab2021 and super parameters in genetic operation are $p = 0.7 - 0.9, m = 0.1 - 0.6, pop_{num} = 8, pop_{size} = 25$. Calculate and visualize the changes of solutions under different objectives and algorithms and the final Gantt chart of workpiece processing sequence, and compare the algorithm performance before and after improvement, including rate of convergence, optimal solution hit rate, CPU usage time and other evaluation indicators.

4.3 Experimental Results and Analysis

The traditional genetic algorithm obtains the optimal processing time of 49s when the genetic parameters remain unchanged. The following figure shows the Gantt chart of the evolution process and arrangement. The blue line represents the fitness value of the best individual in the population, and the red line represents the change in the average individual fitness value of the population:

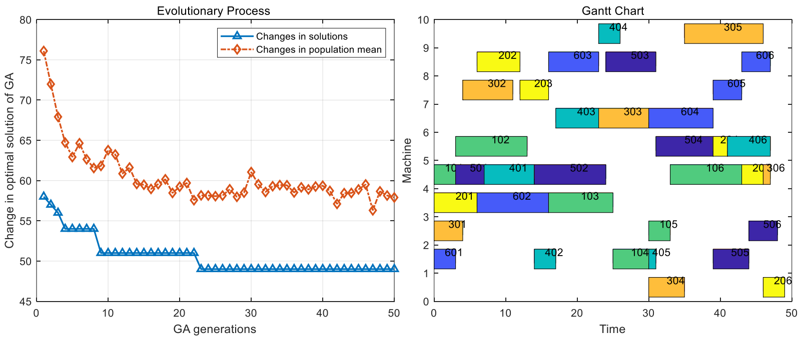

Figure 4 Optimization process of traditional genetic algorithm

From the graph, it can be observed that the optimal solution of the single population genetic algorithm no longer changes around the 22nd generation, but it has not reached the global optimal solution. Due to insufficient individual diversity in the population, the algorithm is prone to premature convergence and falls into local optima.

Improve traditional genetic algorithms by increasing the population size and thus increasing the diversity of solutions. The following figure shows the evolution process and process arrangement Gantt chart of the multi-population genetic algorithm. The optimal processing time is 44s, the dotted line is the average solution of each population, the red solid line represents the change of the multi-population genetic algorithm solution, and the other solid lines represent the change of the single population genetic algorithm solution. It can be seen from the figure below that the multi-population genetic algorithm entered the global optimum in the 11th generation, effectively avoiding the premature of the algorithm, and having a faster rate of convergence. However, due to the expansion of the population, the CPU usage time increased from 0.482 seconds to 3.832 seconds.

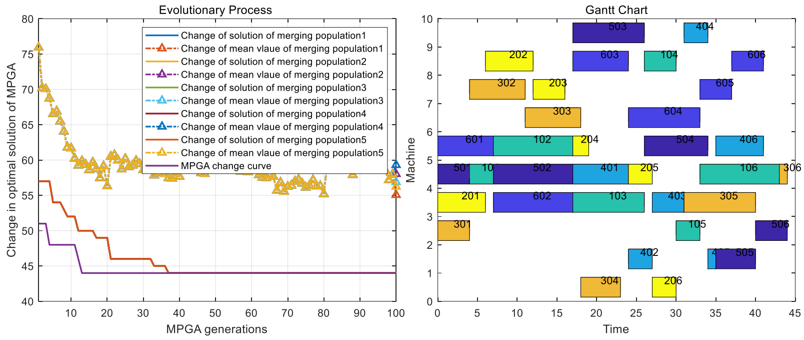

Figure 5 Optimization process of multi-population genetic algorithm

However, in actual production, the delivery time required by customers is sometimes more important than faster processing completion. By introducing fuzzy membership, that is, customer satisfaction as the second objective function, the following process Gantt chart is obtained.

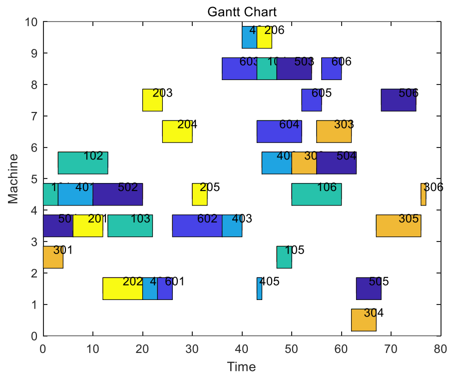

Figure 6 Gantt chart of Multi-objective Multi population Genetic Algorithm

From the above figure, it can be seen that compared to single target tasks, the final completion time of each part is delayed, but the delivery time window that basically meets the customer's requirements is basically met. Table 5 shows the fuzzy satisfaction of customers with each workpiece. Due to the higher set customer satisfaction weight, the completion time of the workpiece has been extended by $\frac{1}{3}$. However, the satisfaction of all 5 customers is 1, and the satisfaction of all 1 customer is 0.8:

Table 5 Customer Fuzzy Satisfaction

workpiece	Submission time	Fuzzy satisfaction
Workpiece 1	59	1
Workpiece 2	51	0.8
Workpiece 3	80	1
Workpiece 4	51	1
Workpiece 5	73	1
Workpiece 6	57	1

Compared with the traditional genetic algorithm, the improved multi-population genetic algorithm has a stronger global search ability, that is, the number of optimal solutions in life and a faster rate of convergence. Compared with traditional multi-population genetic algorithms, their CPU usage time has also significantly decreased. The performance comparison is shown in the table below:

Table 6 Performance Comparison

algorithm	Convergence time (generation)	CPU usage time (seconds)	Global search capability (%)
GA	8	0.482	3
MPGA	45	5.930	38
IMPGA	22	3.832	61

5. Summary

In this paper, the fuzzy objective of customer satisfaction is introduced and the traditional objective and fuzzy objective are weighted into single objective tasks. Based on the traditional job shop scheduling constraints, fuzzy completion time constraints are added to build a fuzzy objective job shop scheduling model. And based on population similarity, a clustering-based population merging strategy is proposed. This algorithm adds immigration operators, merging operators, and manual selection operators to achieve the sharing of excellent genes among individuals. Different immigration and merging cycles are designed to prevent premature convergence in the algorithm. The above experimental simulation verifies that the algorithm has stronger stability: the hit rate of the optimal solution is improved and the Rate of convergence is faster.

However, due to the introduction of multiple populations, the space complexity of the algorithm increases. In the future, researchers can reduce the CPU usage time of the algorithm and improve CPU efficiency by parallelizing the evolution process of each population.

References

[1] Chanas S, Kasperski A. Minimizing maximum latency in a single machine scheduling problem with fuzzy processing time and fuzzy due dates Engineering Appliance of Artistic Intelligence 2001, 14 (3): 377-386

[2] Chanas S, Kasperski A. On two sides machine scheduling problems with fuzzy processing time and fuzzy due dates European Journal of Operational Search, 2003, 17 (2): 281-196

[3] Itoh T, et al Fuzzy due date scheduling problem with fuzzy process time International Transactions in Operations Search, 1999, (6): 639-647

[4] Sakawa M, et al Fuzzy processing for multi_ Objective job shop scheduling with fuzzy processing time and fuzzy schedule through genetic algorithms European Journal of Operational Search, 2000, 120:393-407

[5] Yuan Kun, Zhu Jianying, Wang Xiyang. Fuzzy objective flexible workshop scheduling problem based on genetic algorithm. Mechanical Science and Technology, 2006,25 (10): 1-3

[6] Liu Aijun, Yang Yu, Xing Qingsong, Lu Hui, Zhang Yudong. Multi-population genetic algorithm in multi-objective fuzzy flexible shop scheduling. Computer Integrated Manufacturing System, 2011, 17 (9): 1−5

[7] Yang Fan, Fang Chenggang, Hong Rong, Wu Weiwei. Application of improved genetic algorithm in job shop scheduling. Journal of Nanjing Tech University, 2021,43 (4): 3-5

[8] Zhong Huichao. Research on shop scheduling method based on enhanced genetic algorithm. Huazhong University of Science and Technology, 2019:29-35

[9] Li Zhilin. Research on Genetic Algorithm in Workshop Scheduling Clock. Software Engineering, 25 (10): 1-4

[10] Liu Guanquan, Pan Dandan, Shen Ruqing. Solving Flexible Job Shop Scheduling Problem with AGA-DE. Journal of Lanzhou Institute of technology, 2023, 30(3): 1-3

[11] Guan Sai, Xiong Hegen. Research on hybrid Tabu search genetic algorithm for job shop scheduling. Intelligent Computers and Applications, 2023, 13(5):1-2

[12] Zhou Pengpeng, Zhai Zhibo, Dai Yusen. Research on Flexible Job Shop Scheduling Problem Based on Improved Genetic Algorithm. Modular Machine Tools and Automated Processing Technology, 2022: 1-3

[13] Yu Lei, Shi Feng, Wang Hui, Hu Fei. Analysis of 30 Cases of MATLAB Intelligent Algorithms. Beihang University Press, 2015: 108-117