Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on Integrated Process Planning and Scheduling Based on Interval Theory

Fang Han[1,✉], Lijun Liu[1]

[1] *School of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an, Shaanxi, 710021, China*

ABSTRACT

This paper studies integrated process planning and scheduling (IPPS), a typical workshop scheduling problem, and mainly investigates the uncertain problems in the actual industrial production process. Then, we introduce the theoretical knowledge of interval numbers and adopt the interval number comparison method. Specifically, interval numbers are used to replace the determined processing time, and uncertain IPPS problems are modeled based on the interval number theory. Based on this, a hybrid particle swarm algorithm is proposed to solve the uncertain IPPS. Meanwhile, the genetic operator is introduced to improve its ability to deal with combined optimization problems. The above theoretical results are applied to the process planning and scheduling of a mechanical workshop, thus verifying the effectiveness of the proposed method.

Keywords: process planning, workshop scheduling, uncertain problem, particle swarm optimization

1. Introduction

In traditional enterprise production and manufacturing systems, workshop scheduling is always carried out separately after the process planning is determined, instead of being considered coordinately. Research on IPPS can greatly improve the production efficiency of a workshop, and provide theoretical and application guidance for the actual production system. Currently, factors that frequently occur in workshop production, such as the arrival of new orders, the influx of urgent orders, and changes in order delivery deadlines, are not taken into account. Therefore, an uncertainty analysis of IPPS issues will help to organically unify process planning and production scheduling, improve production efficiency, reduce production costs, shorten production cycles, enhance the competitiveness, and create more profits for enterprises.

In the actual production process, the processing environment will be affected and changed by external factors such as the suspension of production due to equipment trouble and uncertain

✉ Corresponding author.
E-mail address: littlefang2025@163.com (F. Han).
Received 12 January 2024; Revised 25 March 2024; Accepted 25 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-410

processing time. As uncertainty research matures, uncertainty solutions are frequently applied in optimizing scheduling problems, and uncertainty factors in workshop scheduling have become the research focus and hot topic today. For the handling of uncertain events of equipment failure, rolling window technology and event-driven rescheduling strategies are generally used. For the uncertain processing time, fuzzy numbers, interval numbers, and random numbers are often adopted to represent it in existing studies.

1) Research status of uncertain processing time. On using fuzzy numbers to describe the workshop scheduling problem of uncertain processing time, Nezhad et al. [1] introduced an appropriate approximation for the maximum operator, proposed to schedule the processing sequence based on the preference ratio concept, and developed a novel fuzzy CDS algorithm for fuzzy flow workshop scheduling. Li et al. [2] proposed a three-goal model to solve the MOIPPS problem which used six-layer coding, semi-active decoding, hierarchical selection and tournament selection. For mutation, a combination of Swap and single point was used to construct a Pareto solution set. Hu et al. [3] studied the workshop scheduling problems of fuzzy processing time and fuzzy delivery dates. They introduced the concept of fuzzy number sequencing based on the measurement of probability and necessity, and would design an improved DE algorithm to solve these target functions. Chen et al. [4] offered a fuzzy number scheduling model, separated the process from the machine, carried out genetic operations separately, and replicated using the gamble wheel method.

On using interval numbers to describe the workshop scheduling problem of uncertain processing time, Lei [5] applied the interval number theory to production scheduling, gave full play to its advantages in uncertainty modeling, and proposed the population-based neighborhood search (PNS) method to optimize the interval makespan of the problem. The arithmetic structure of interval numbers was utilized to decode the process. Matsveichuk et al. [6] used the interval number to solve the workshop scheduling problem of parallel flow. Yang et al. [7] addressed the JSP-PTV/DDW problem, transformed it into a range problem, and made a theoretical proof. They added elite cross to the algorithm, and used Range operations to perform semi-active decoding and adaptive value comparison. Peng et al. [8] designed a model for solving rescheduling with gray numbers. They used the conversion relationship between the interval numbers and the gray numbers, adopted the arena's principle of gray number sequencing, and also constructed a set of non-key processes to perform heuristic scheduling. Chen [9] obtained the initial group by range selection for the three-goal problem under uncertain working hours. In addition, IPOX was used to cross and mutant the parent generation, so as to obtain a new child. Pareto sorting and crowding distance calculation were adopted to select the child generation, making the NSGA-II method more suitable for solving this problem.

On using random numbers to describe the uncertain processing time, Jia [10] conducted a study on the stand-alone random scheduling problem where the processing time followed an exponential distribution, in order to minimize the processing time of workpieces. Manna DK et al. [11] used the random number theory to discuss the stand-alone delivery schedule problem where the processing time satisfied the exponential distribution. Jia C [12] discussed the stand-alone scheduling problem where the manufacturing time was uncertain, so as to minimize the completion time and random deadline. The processing time and deadlines were distributed exponentially.

2) Research status of workshop scheduling under machine failure. In order to study JSP, Wang [13] established a model where two common interference factors in the workshop were chosen. Based on the range processing data and random machine failure, he established three optimization algorithms to improve and adjust the feeding order. The results were used in Plant Simulation. After the processing process was simulated, the obtained target values were used in GA's subsequent calculations. Jamal et al. [14] studied the DJSS problem where the arrival of workpiece was random and the machine had failed, and proposed a rescheduling method for variable neighborhood search.

Reinforcement learning and the Q-factor algorithm were used to improve the performance of the scheduling method. Sobaszek et al. [15] explained the impact of machine failure on the processing completion time, and used the time prediction algorithm to solve the problem. To solve the machine malfunction in the workshop, Tang et al. [16] simulated the fault situation, determined whether to start rescheduling based on whether the fault affects processing, and then solved the problem through the simulation of the actual situation and algorithm simulation. Ba [17] provided a new method for describing the relationship among processes, and also gave a plan to evaluate the scheduling plan. Finally, he constructed a multi-objective model and improved GA for problem solving. Liu et al. [18] analyzed the dynamic constraints at the time of failure, constructed a set of dynamic constraints and tasks, and used NCL modeling to minimize the total delay time after the failure occurred.

To sum up, there are more examples of using fuzzy numbers and interval numbers to represent uncertain processing time. A few papers also use the random number theory. It is shown that for fuzzy numbers or random numbers, a large number of data are needed to determine their distribution. As a special case of fuzzy numbers, the interval number reveals the interval where the data are located, not the interval where the data must meet a specific probability. In this case, its affiliation function structure is not complicated.

Based on the research on traditional integrated process planning and workshop scheduling problems, this paper uses the interval number to describe the uncertain processing time, so as to simplify calculation and solve the problem more efficiently. In order to better simulate the actual processing environment, we take into account the uncertainty of processing time and try to minimize the maximum completion time. Then, a scheduling model with interval processing time is created, and a hybrid optimization algorithm based on particle swarm optimization is proposed. When particles are looking for feasible solutions, they may be guided by unilateral information and approach to the local optimal position, making them difficult to jump out of this constraint. Therefore, combining the two algorithms can improve the single algorithm and draw on each other's strengths.

2. Uncertain IPPS problem model based on the interval theory

2.1. Description of the problem

IPPS problem: there are n workpieces to be processed, and there are m machines in the workshop. Each part is processed through multiple processes which can form multiple process routes in different orders. Each process can be completed by one or more optional machines. According to the actual situation, the following assumptions are made for uncertain production process planning and workshop scheduling problems.

1) There is no sequential constraint between workpieces and they are independent of each other;
2) Each machine can only process one process at one time;
3) Different processes of the same workpiece cannot be processed at the same time;
4) Each process of each workpiece cannot be stopped once it is started.

Based on the above assumptions, we list the relevant parameters and their definitions. See Table 1.

Table 1. Parameters and their definitions.

No.	Parameter	Definition
1	O_{ijl}	the *j*th process of the *i*th workpiece and the *l*th optional process route selected for the workpiece.
2	k	the corresponding optional processing machine of O_{ijl}.
3	G_i	the number of optional process routes of the *i*th workpiece.
4	P_{il}	the total number of operations included in the *l*th optional process of the *i*th workpiece.
5	$\%C_{ijkl}$	the completion time of the uncertain interval of the process.
6	$\%Y_W(i,j,k,l)$	the uncertain interval processing time required on machine k for O_{ijl}.
7	$\%Y_S(i,j,k,l)$	the earliest uncertain interval start-up time on machine k for O_{ijl}.
8	$\%Y_E(i,j,k,l)$	the interval completion time required on machine k for O_{ijl}.
9	X_{il}	When the *l*th process route is selected for workpiece *i*, its value is 1; otherwise, it is 0.
10	Z_{ijkl}	When the process O_{ijl} is processed on machine k, its value is 1; otherwise, it is 0.

2.2. Mathematical models

$$\text{Min} \quad Makespan = Max\left\{\%C_{ijlk}\right\} \tag{1}$$

s.t.:

$$\sum_l X_{il}=1, \quad X \in [1, G_i] \tag{2}$$

$$\sum_{k=1}^{M} Z_{ijlk}=1, \quad \mathrm{i} \in [1,\mathrm{N}], \quad \mathrm{j} \in [1,\mathrm{P}_{il}], \quad l \in [1, G_l] \tag{3}$$

$$\%Y_E(i,j,k,l)=\%Y_S(i,j,k,l)+\%Y_W(i,j,k,l) \tag{4}$$

$$\%Y_S(i,j,k,l) \geq \%Y_E(i,j,k,l) \tag{5}$$

$$\begin{aligned} &\%Y_S(i,(j+1),k_2,l)-\%Y_S(i,j,k_1,l) \geq \%Y_W(i,j,k_1,l) \\ &i \in [1,N], \quad \mathrm{j} \in [1,\mathrm{P}_{il}], \quad l \in [1,G_l] \end{aligned} \tag{6}$$

$$\begin{aligned} &\%Y_S(i,j_2,k,l)-\%Y_S(i,j_1,k,l) \geq \%Y_W(i,j_1,k,l) \\ &i \in [1,N], \quad j_1, j_2 \in [1,\mathrm{P}_{il}], \quad l \in [1,G_l] \end{aligned} \tag{7}$$

Eq. (1) defines the target function for minimizing the maximum completion time. Eq. (2) indicates that only one process route can be selected for one workpiece. Eq. (3) shows that only one optional processing machine can be selected in one process. Eq. (4) means that the completion time of the workpiece process is equal to the processing starting time plus the processing time. Eq. (5) indicates the sequential constrained relationship of all processes of the same workpiece. In Eq. (6), each machine can only process one process at the same time. Eq. (7) shows that one workpiece can only be processed by one machine at the same time. In the above equations, the processing time is modeled as an interval number. The data are divided into different intervals according to Ghrayeb's method: Given a determined processing time *x*, the number of intervals will be $\delta_1 x$, $\delta_2 x$, of which $\delta_1 < 1, \delta_2 > 1$. The two parameters refer to the possibility of early completion and late completion. In fact, the probability of a process being completed late is always greater than that of early completion. Thus, $\delta_1=0.85, \delta_2=1.3$.

3. Method for solving the uncertain IPPS problem based on genetic algorithms

3.1. Encoding, decoding and initialization methods

3.1.1. Encoding. The combined coding method is adopted, which is a scheme of coding by traversing the process routes and the sequence of assigned machines. Each solution is actually the permutation and combination of the process routes and the scheduling plans that meet the constraints. Coding is to assign each process and machine of each process as genes on chromosome, and these genes can be arranged and combined to obtain different coding schemes.

3.1.2. Decoding. Decoding requires reverse thinking of the coding process. First, the processing machines assigned to each process must be clearly defined according to the chromosome number, and then the processing sequence of each process on each machine can be determined.

3.1.3. Initialization method. This section uses a random initialization method. N initial strings of structured data are randomly generated. Each string of structured data is called an individual, and N individuals form a group. The evolution begins with these N strings of structured data as the initial point.

3.1.4. Fitness function. Genetic algorithms use fitness values as a measure of current solutions. The fitness value of each individual reflects the fitness which is used to guide the search. For different problems, different fitness functions should be established to evaluate the population, which is closer to the real problem. In this paper, the objective function is taken as the fitness function.

3.2. Tournament selection strategy

Individuals are selected based on their fitness with the principle of survival of the fittest. That is, high-quality individuals in the population are found, so that their good performance can be preserved and then passed on to offspring with a higher probability. It improves the quality of the population. Similarly, the optimal solution to the algorithm can also be found faster.

This paper uses binary tournament selection. Two individuals are taken from the population at once, and the best one is selected from the two individuals and retained in the next generation of the population set. The greater the fitness value, the greater the probability of being selected. This operation is repeated several times until the number of selected individuals is equal to the original population size.

3.3. Crossover and mutation design

3.3.1. Crossover operation. Crossover operation refers to the exchange of some genes from two parent individuals in a certain manner, so as to obtain the children with better performance. The essence is to recombine the excellent genes from the two parent chromosomes to ensure genetic diversity in the offspring.

Unlike the crossover and mutation of traditional genetic algorithms, we propose an improved genetic operation. First, set a crossover probability p, and randomly select a point $G1$ in the chromosome, such as $G1=1$. Then, generate a random decimal. If it is less than p, the second point $G2$ comes from another arbitrary point of the same individual, such as $G2=6$. Further, invert the part between $G1$ and $G2$ to get a new individual, as shown in Figure 1:

If the random decimal is greater than *p*, select an individual from the population at random, and find the point which locates at the previous position of $G1=1$ in the individual. If the previous point $G3=10$, return to the original individual, and invert the numbers between 1 and 10, as shown in

Figure 2. This genetic method can use the obtained population information as much as possible to guide the cross-update operation of individuals, making genetic operators more efficient.

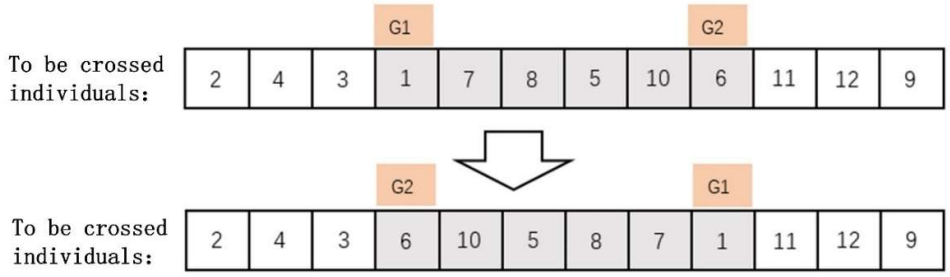

Figure 1 Crossover operation when $q < p$

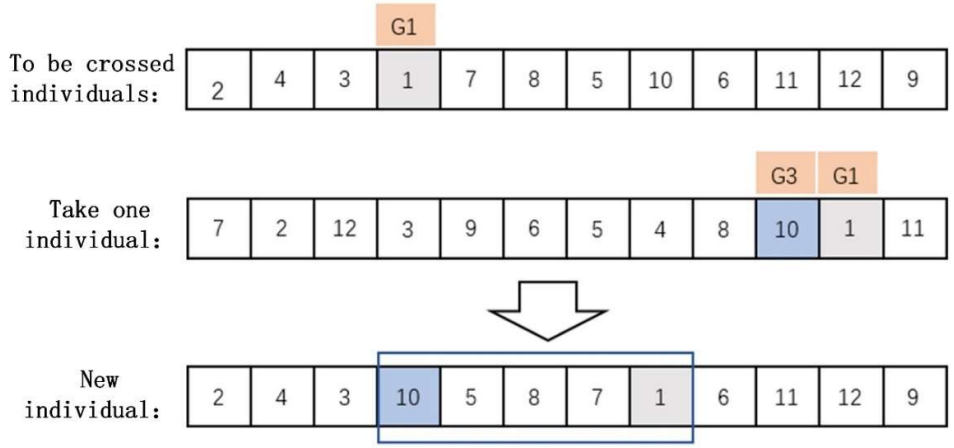

Fig. 2. Crossover operation when $q \geq p$

3.3.2. Mutation operation. First, set the mutation probability and then select an individual randomly from the group. For the selected individual, select two genetic locuses $G1$ and $G2$ randomly, exchange the information carried by them, and place the parts of them reversely to generate offspring generations. Each gene locus has the same probability to be exchanged with any gene locus to complete the mutation. This strategy can minimize the perturbation degree once, which is beneficial to local search, as shown in Figure 3.

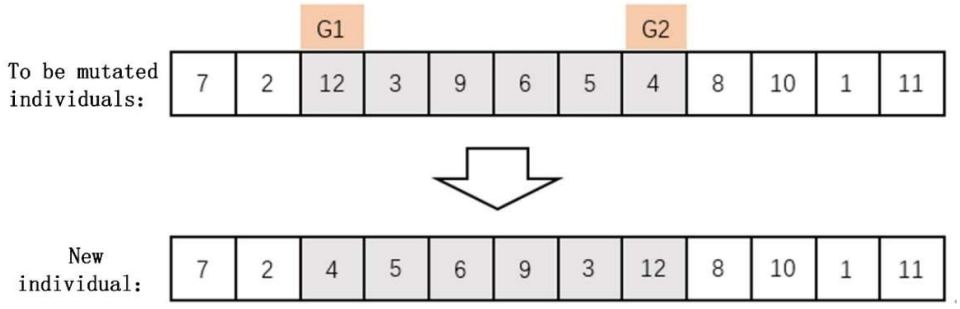

Fig. 3. Mutation operation

3.4. Algorithm process

1) Determine parameters such as population size, number of iterations and genetic operator probability;

2) Initialize the population;
3) Calculate the population fitness;
4) Generate the set of optimal individuals for each generation;
5) Generate the set of individuals with the highest fitness for each generation;
6) Complete the select operation and adopt the binary tournament strategy;
7) Conduct the crossover and mutation operations;
8) Calculate the adaptability of the new population;
9) Combine child and parent populations and complete the selection;
10) Return to the final evolved population.
Figure 4 shows the flow chart of the algorithm.

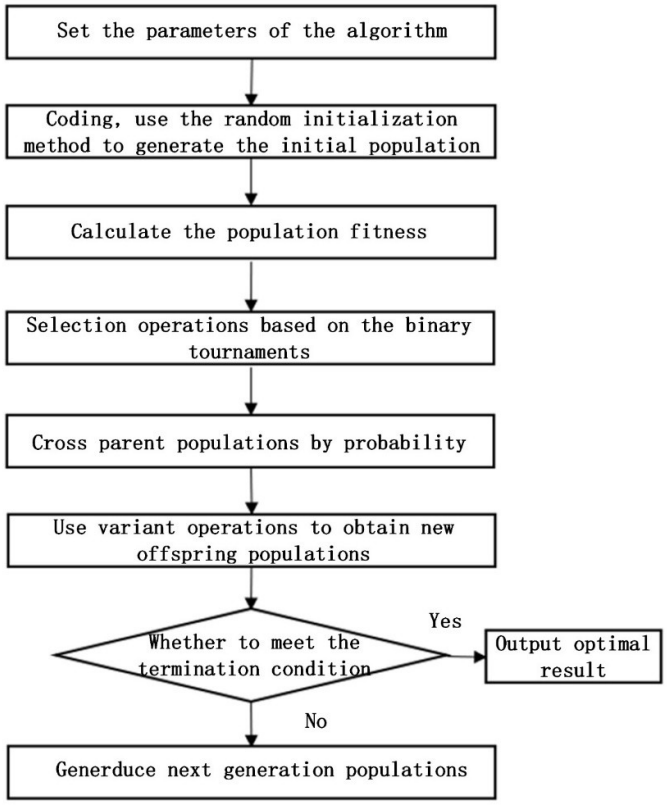

Fig. 4. Flowchart of the genetic algorithm

3.5. Case analysis

3.5.1. Case process information. The manufacturing shop of a parts manufacturing machinery factory is selected as the research object. There are 9 devices in the workshop. In the actual manufacturing process, 12 kinds of parts are produced in the workshop at the same time. Table 2 gives standard working hours, and the data will be intervalized during the practical solution. They are solved by the genetic algorithm and particle swarm optimization, respectively. Eq. (1) in Section 2 is the optimization target function.

Table 2. Processing information.

	Workpiece	Process								
		1			2			3		
Processing time	1	18	21	23	15	30	27	31	25	29
	2	9	12	15	7	11	18	16	6	9
	3	13	18	25	14	43	40	39	55	60
	4	26	33	29	28	21	20	16	17	17
	5	4	3	4	6	8	7	5	9	4
	6	24	18	22	25	21	18	20	31	26
	7	10	13	15	8	2	7	8	6	5
	8	27	30	25	26	34	39	32	8	10
	9	19	25	23	25	21	26	21	16	13
	10	9	14	6	11	19	24	23	8	5
	11	39	34	41	38	27	31	25	20	30
	12	20	18	18	22	5	7	4	16	12

We set the population size (PopulationSize) as 20, the maximum number of evolutionary generations (maxGeneration) as 200, the crossover rate (crossoverRate) as 0.6, and the mutation rate (mutationRate) as 0.01. Matlab R2017b is used to run the algorithm and test data, including 12 workpieces and 9 machines. The evolution curve of the population obtained by applying the genetic algorithm and the Gantt chart for the optimal solution are shown in Figure 5-Figure 7. Table 3 shows the optimal solution, the worst solution and the average solution to the ten optimal processing results of the case.

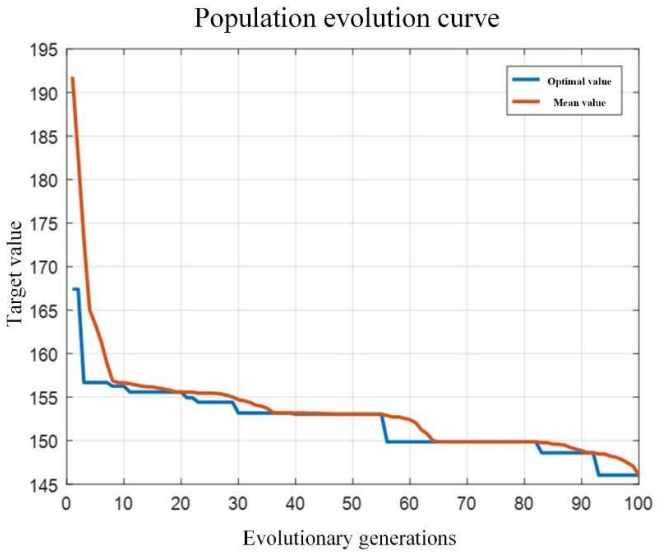

Fig. 5. Population evolution curve (100 generations)

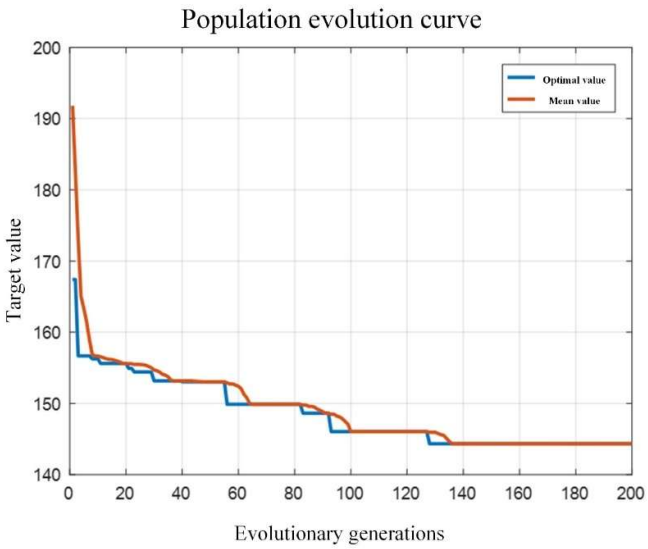

Fig. 6. Population evolution curve (200 generations)

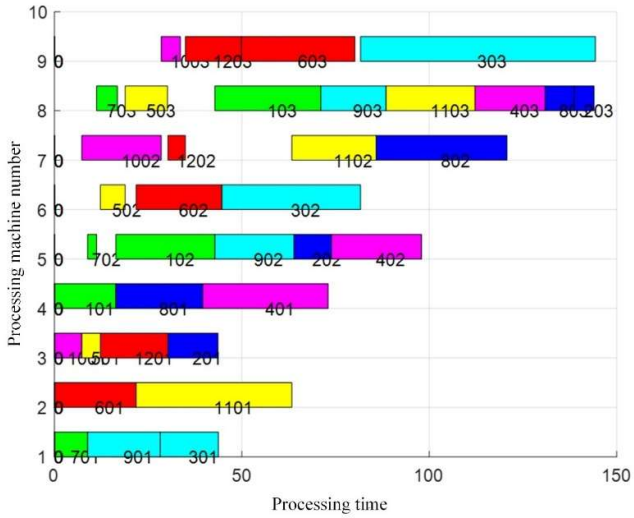

Fig. 7. The optimal scheduling scheme

Table 3. Calculation results of the case.

Number of runs	Results (s)
1	152.274
2	154.950
3	149.835
4	149.125
5	151.871
6	152.548
7	151.528
8	144.330
9	159.016
10	155.892
mean	152.137

Figure 6 shows that the algorithm obtains the optimal solution after 140 iterations, and the convergence speed is slow. In Figure 7, the ordinate represents the serial number of the processing machine while the abscissa represents the processing time. The serial number denotes the serial number of the workpiece where the first two digits from left to right represent the workpiece number and the last two digits represent the process number. As can be seen from Table 5, the obtained ten optimal processing results are not stable. The optimal solution is 144.330s, the worst is 159.016s, and the average is 152.137s. It is worth noting that as the scale of the problem increases, the optimization ability of the algorithm has declined. The reason is that when crossover and mutation calculation are conducted, no comprehensive directional optimization is achieved by GA algorithm according to its iteration rules. Therefore, the algorithm needs to be improved to increase the directional optimization ability in large-scale problem spaces.

4. Method for solving uncertain IPPS problems based on the GA-PSO hybrid algorithm

4.1. Coding, decoding and initialization methods

4.1.1. Encoding. Combined coding method is adopted in this section, a scheme of coding by traversing the process routes and the sequence of assigned machines. Each solution is actually the permutation and combination of the process routes and the scheduling plans that meet the constraints. Coding is to assign each process and machine of each process as genes on chromosome, and these genes can be arranged and combined to obtain different coding schemes. Specifically, the sequence of processes is arranged first, and the assignment of machines, are considered. Then, all possible situations are listed, and finally we eliminate the combinations that do not meet the constraints.

4.1.2. Decoding. Decoding requires reverse thinking of the coding process. First, the processing machines assigned to each process must be clearly defined according to the chromosome number. Then the processing sequence of each process on each machine can be determined.

4.1.3. Initialization method. This section uses a random initialization method. N initial strings of structured data are randomly generated. Each string of structured data is called an individual, and N individuals form a group. The evolution begins with these N strings of structured data as the initial point.

4.2. Elite selection strategy based on interval numbers

In the improved hybrid algorithm in this paper, selection is performed by selecting the best individual based on the interval number. The process is:

1) The best individuals (1%) are selected from the parent population according to the comparison rules for the probability of intervals, and then directly copied to the next generation. The chromosomes are sorted according to the fitness value.

2) When a selection is completed, put the selected individuals back into the population. Also, the selected individuals can continue to participate in selection as the parent chromosome. It aims to ensure that all the selected individuals will not be damaged by subsequent crossover and mutation operations, thus improving the reliability of simulated convergence. Specifically, we sort the completion time of workpieces through the comparison rule of the interval number, and choose processing machines according to the process that takes less time. Considering that the processing time of uncertain processes in this paper are all interval numbers, the interval number comparison method based on probability proposed is used, as shown in Eq. (8). There are two proper scheduling schemes $\tilde{A}=[A^-, A^+]$ and $\tilde{B}=[B^-, B^+]$. Then:

$$P(A \le B) = \begin{cases} 0 & ,A^- \ge B^+ \\ 0.5 \times \dfrac{B^+ - A^-}{A^+ - A^-} \times \dfrac{B^+ - A^-}{B^+ - B^-} & ,B^- \le A^- < B^+ \le A^+ \\ \dfrac{B^- - A^-}{A^+ - A^-} + 0.5\dfrac{B^+ - B^-}{A^+ - B^-} & ,A^- < B^- < B^+ \le A^+ \\ \dfrac{B^- - A^-}{A^+ - A^-} + \dfrac{A^+ - B^-}{A^+ - A^-} \times \dfrac{B^+ - A^+}{B^+ - B^-} & \\ +0.5 \times \dfrac{A^+ - B^-}{A^+ - A^-} \times \dfrac{A^+ - B^-}{B^+ - B^-} & ,A^- < B^- \le A^+ < B^+ \\ \dfrac{B^+ - A^+}{B^+ - B^-} + 0.5 \times \dfrac{A^+ - A^-}{B^+ - B^-} & ,B^- \le A^- < A^+ < B^+ \\ 1 & ,A^+ \le B^- \end{cases} \tag{8}$$

4.3. Crossover and mutation operation based on the core principles of the particle swarm algorithm

The particles move according to the updated formula of position and speed. Then, select the current optimal particle, the globally optimal particle and the historically optimal particle from the generated particles and combine one with each other. The corresponding disturbance operation is carried out with the set probability of crossover and mutation. That is, each chromosome is crossed with the current optimal, the globally optimal, and the historically optimal individual. Then, mutation operation is conducted on it. Finally, select the one with the best fitness from the three chromosomes and put it into the next generation.

4.3.1. Particle movement principle. The updated equations of particles' velocity and position are shown in Eq. (9) and Eq. (10), respectively:

$$v_i^d = wv_i^{d-1} + c_1 r_1 (pbest_i^d - x_i^d) + c_2 r_2 (gbest^d - x_i^d) \tag{9}$$

$$x_i^{d+1} = x_i^d + v_i^d \tag{10}$$

Eq. (9) indicates that the particle's velocity at step *d* is equal to the inertia of the speed at the previous step + the self-perception part + social perception part. Eq. (10) indicates that the particle's position at step *d+1* = the position where the particle is at step *d* + the position where the particle is at step *d* * the time of movement (the time *t* for each step is generally taken as 1). r_1 and r_2 are generally any number between (0, 1). c_1 is the individual learning factor of the particle. c_2 indicates the social learning factor of the particle. w represents the inertia weight of the speed. v_i^d is the velocity of the *i*th particle in the *d*th iteration. x_i^d is the position of the *i*th particle in the *d*th iteration. $pbest_i^d$ shows the best position traversed by the *i*th particle until the *d*th iteration. $gbest_i^d$ is the best position for all particles to traverse up to the *d*th iteration. The specific search principle of particles is shown in Figure 8.

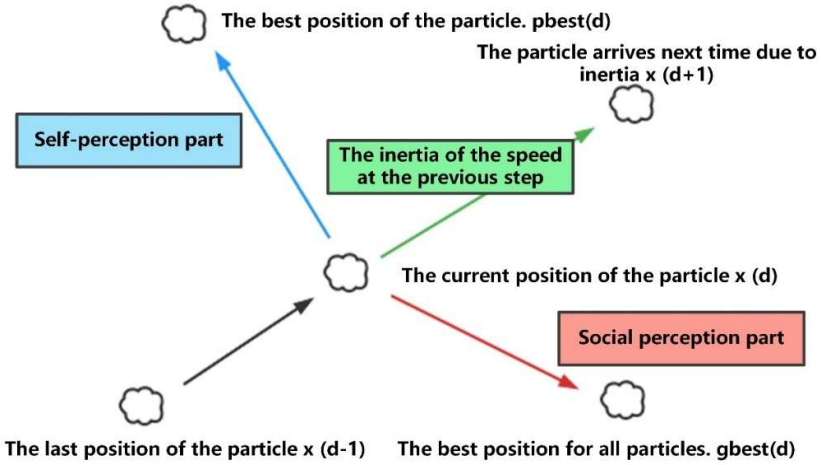

Fig. 8. Particle swarm algorithm search strategy

4.3.2. Crossover operation. Unlike the crossover and mutation methods of traditional genetic algorithms, we propose an improved genetic operation. First set a crossover probability p, and randomly select a point $G1$ in the chromosome, such as $G1=1$. Generate a random decimal q between [0,1]. If the number is less than p, the second point $G2$ comes from another arbitrary point in the same individual, such as $G2=6$. Then, invert the part between points $G1$ and $G2$ to get a new individual, as shown in Figure 9. If q is greater than p, select an individual from the population at random, and find the point which locates at the previous position of $G1=1$ in the individual. If the previous point $G3=10$, return to the original individual and invert the numbers between 1 and 10, as shown in Figure 10.

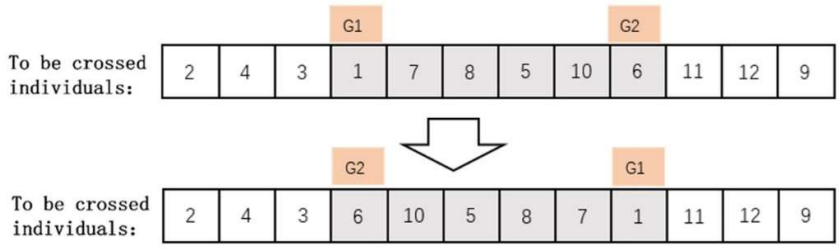

Fig. 9. Crossover operation when $q < p$

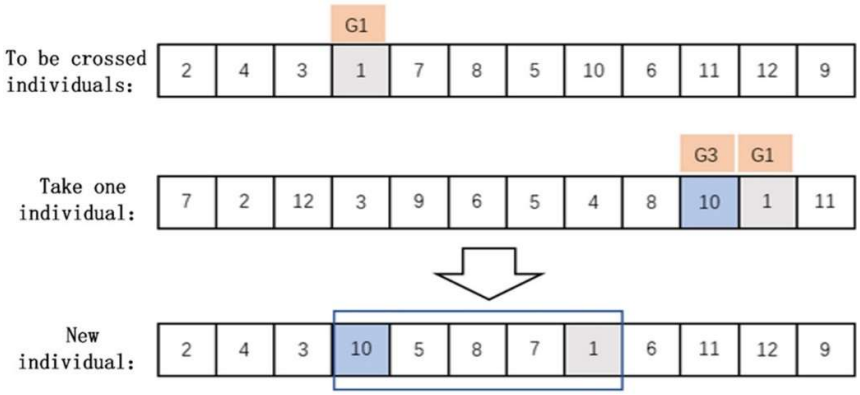

Fig. 10. Crossover operation when $q \geq p$

4.3.3. Mutation operation. First, set a mutation probability, and then select an individual at random from the group. For the selected individual, select two genetic locuses G1 and G2 randomly, exchange the information carried by them, and place the parts between them reversely. Each gene locus has the same probability to be exchanged with any gene locus to complete the mutation, as shown in Figure 11.

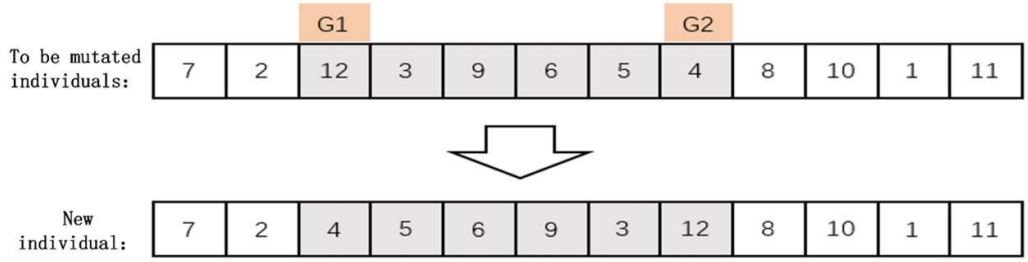

Fig. 11. Mutation operation

4.4. Flow of the hybrid algorithm

1) Set parameters, including population size, iterations, particle velocity range, inertia weight, learning factor, mutation and crossover probability;

2) Initialize the population, generate particles with the size of the population, and assign the generated population to the chromosome of the genetic algorithm;

3) Calculate the fitness value of the initialized population and record the optimal location and optimal chromosome of individuals and populations;

4) Update the speed and position of the particles according to the optimal values of the previous generation population, calculate the fitness value of the updated particle swarm, and keep the optimal value;

5) Perform selection, crossover, and mutation operations. If it evolves to a better individual, replace the corresponding individual and update the individual optimal fitness value or the global optimal value;

6) Judge whether the maximum number of iterations is reached. If so, the algorithm stops. If not, go to Step 4;

7) When the evolution stops, we can obtain the optimal plan and draw a Gantt chart.

The flow chart is shown in Figure 12.

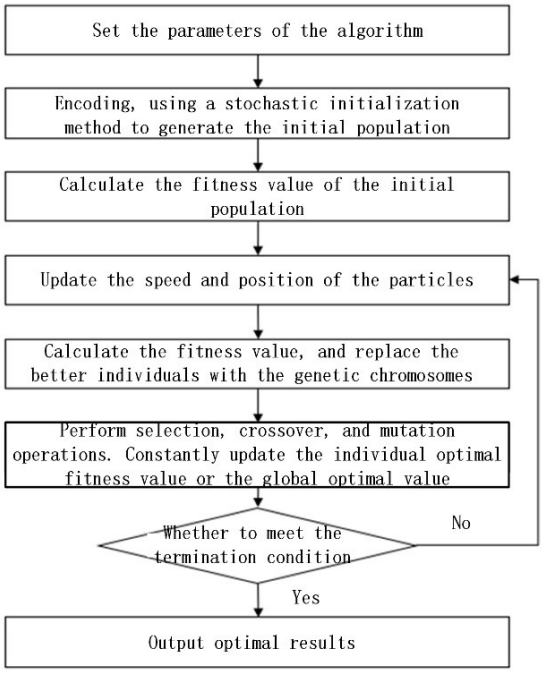

Fig. 12. Flow chart of hybrid algorithm design

5. Case analysis

5.1. Case process information

This section selects the manufacturing shop of a parts manufacturing machinery factory as the research object. There are 9 devices in the workshop. In the actual manufacturing process, 12 kinds of parts are produced in the workshop at the same time. Table 4 gives the standard working hours, and the data will be intervalized during practical solving. They are solved through the genetic algorithm and particle swarm optimization, respectively. Eq. (1) in Section 2 is the optimization target function.

Table 4. Processing information.

	Workpiece	Process								
		1			2			3		
Processing time	1	18	21	23	15	30	27	31	25	29
	2	9	12	15	7	11	18	16	6	9
	3	13	18	25	14	43	40	39	55	60
	4	26	33	29	28	21	20	16	17	17
	5	4	3	4	6	8	7	5	9	4
	6	24	18	22	25	21	18	20	31	26
	7	10	13	15	8	2	7	8	6	5
	8	27	30	25	26	34	39	32	8	10
	9	19	25	23	25	21	26	21	16	13
	10	9	14	6	11	19	24	23	8	5
	11	39	34	41	38	27	31	25	20	30
	12	20	18	18	22	5	7	4	16	12

5.2. Experimental results and analysis

For parameter configuration, we set the population size as 50, the maximum number of evolutionary generations as 200, the crossover probability as 0.6, and the mutation probability as 0.01. The algorithm is run with Matlab R2017b. The evolution curve and optimal solution of the population obtained by applying the our improved particle swarm genetic hybrid algorithm are shown in Figure 13-Figure 18. The population evolution curve obtained after solving the genetic algorithm and the Gantt chart for the optimal solution are shown in the figures below. The obtained ten optimal processing results, including the optimal solution, the worst solution, and the average solution, are compared with the GA solutions. See Table 5.

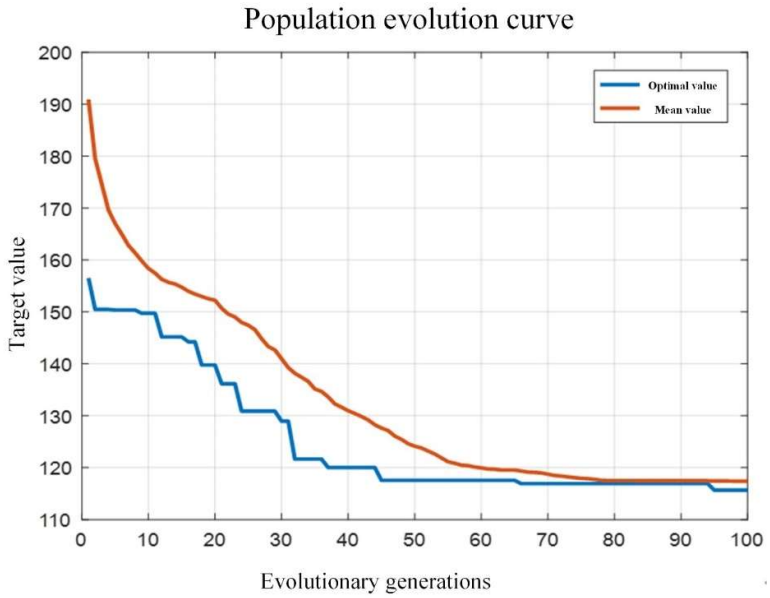

Fig. 13. Population evolution curve of the hybrid algorithm (100 generations)

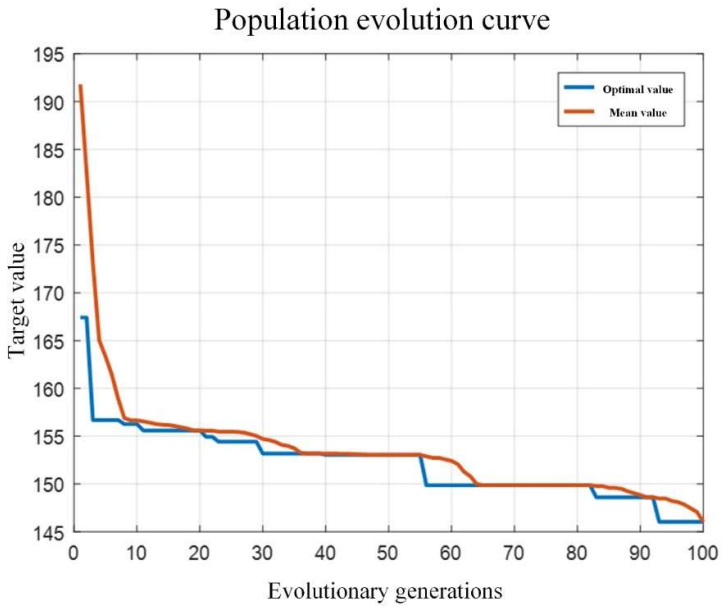

Fig. 14. Population evolution curve of the genetic algorithm (100 generations)

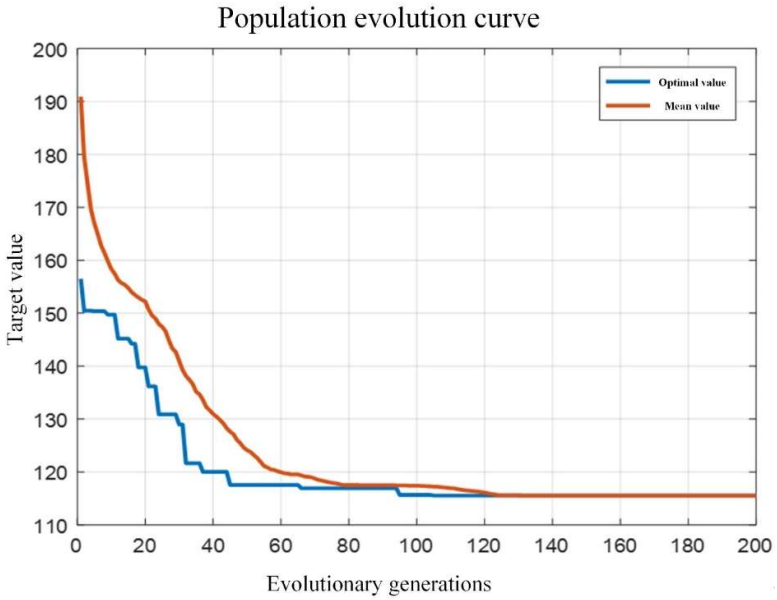

Fig. 15. Population evolution curve of the hybrid algorithm (200 generations)

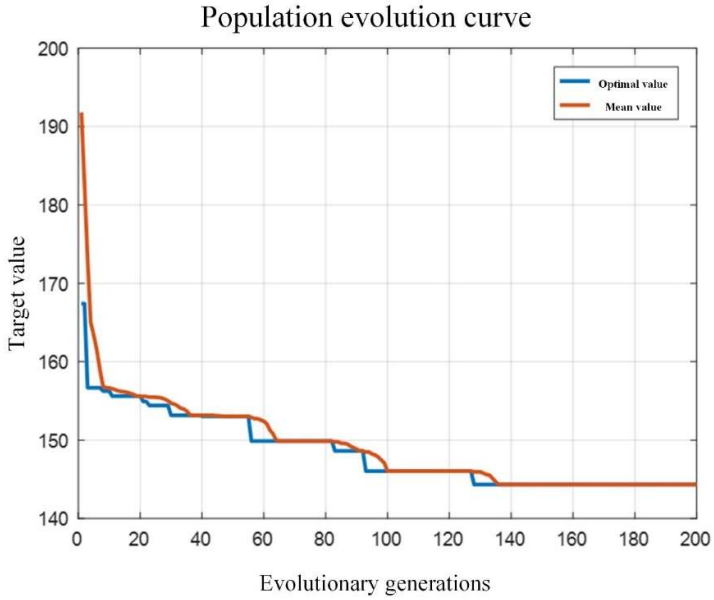

Fig. 16. Population evolution curve of the genetic algorithm (200 generations)

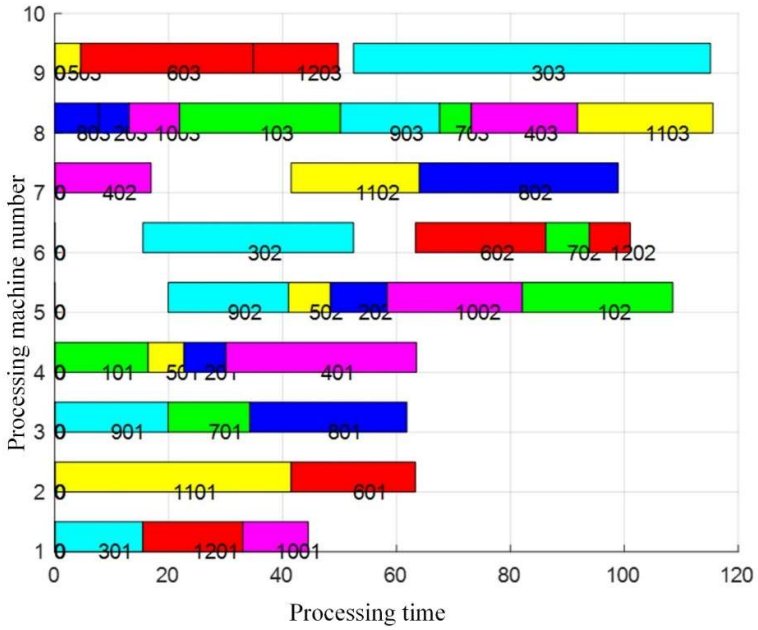

Fig. 17. Gantt chart for the optimal solution to the hybrid algorithm

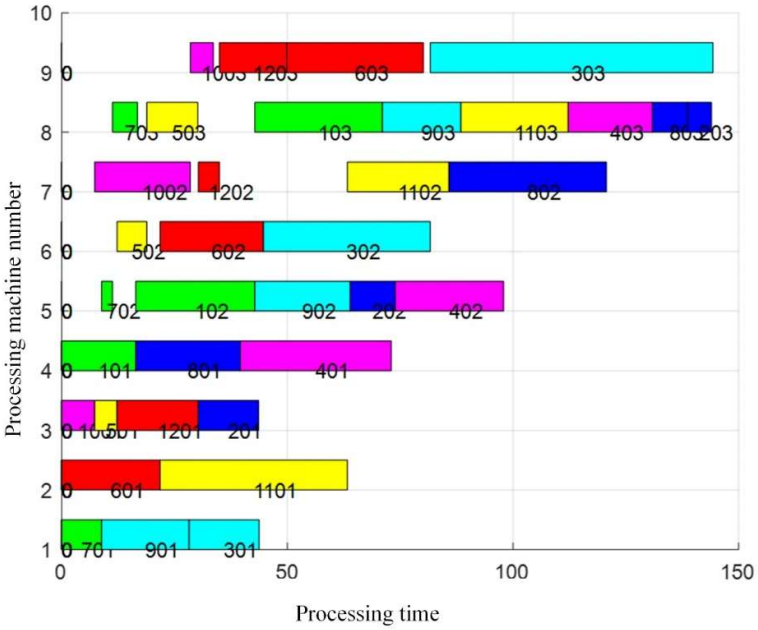

Fig. 18. Gantt chart for the optimal solution to the genetic algorithm

Table 5. Comparison of the statistical results of the two algorithms.

Number of operations	1	2	3	4	5	6	7	8	9	10	mean
Optimal solution to the genetic algorithm	152	155	150	149	152	153	152	144	159	156	152.2
Optimal solution to the hybrid algorithm	117	120	124	130	120	115	118	120	122	126	121.2

Table 5 shows that the results of the genetic algorithm don't converge until the 140th generation. For the GA-PSO hybrid algorithm (composed of the genetic algorithm and particle swarm optimization), the results converge to the optimal solution after the 120th generation which is greatly superior to that of the genetic algorithm. From the Gantt chart, it can be seen that the arrangement of the hybrid algorithm is close and uniform, and the processing schedule on each

device is much more saturated, greatly improving the device utilization rate. The optimal process route for workpieces 1-12 is shown in Figure 16. Table 5 also shows that the optimal maximum processing time obtained by the hybrid algorithm can be minimized to 115/s, which is obviously better than that of the genetic algorithm.

In summary, the improved GA-PSO hybrid algorithm proposed in this paper has better performance in solving uncertain IPPS problems. Its accuracy and convergence speed are superior to those of a single standard genetic algorithm, and can provide enterprise decision makers with a set of proper process planning and workshop scheduling plans. For actual workshop problems, the scheduling problem is more complicated. When the processing time becomes longer, the improved particle swarm hybrid and optimization algorithm can be used to obtain the smaller completion time, greatly reducing the manufacturing time and cost.

6. Conclusion

Based on traditional static manufacturing environment researches, this paper conducts in-depth theoretical research and applied research on IPPS under uncertain disturbances with a focus on uncertain production process planning and workshop scheduling issues. It proposes a method to describe uncertain processing time with the interval number theory, and conducts a systematic study on single-target scheduling problems with an improved particle swarm hybrid algorithm. The results of the study are as follows:

1) In order to better simulate the actual processing environment, we take into account the uncertainty of processing time, combine the interval number theory with the IPPS problems, and create a scheduling model that considers the interval processing time, so as to minimize the maximum completion time.

2) Based on the interval theory and current researches, we apply the probability-based comparison method of the interval number to the selection operation of our algorithm. In this way, interval numbers can be compared more accurately, and all selected individuals will not be destroyed by the subsequent crossover and mutation operations, thus improving the reliability of case simulation convergence.

3) We combine a single particle swarm algorithm with the genetic algorithm, optimize the scheduling scheme to a certain extent, and neutralize the shortcomings of the single algorithm. Therefore, our hybrid algorithm can achieve excellent performance in terms of calculation speed and accuracy.

Acknowledgements

The authors acknowledge the Shaanxi Province Key R&D Plan Project (NO.2019GY-024).

References

[1] Nezhad S S,Assadi R G.Preference ratio-based maximum operator approximation and its application in fuzzy flow shop scheduling[J]. Applied soft computing, 2008, 8(1): 759-766.

[2] Li Yan , Ba Li, Cao Yuan ,Liu Yong,Yang Ming Shun. Research on Integrated Process Planning and Scheduling Problem with Consideration of Multi-objectives [J].China Mechanical Engineering, 2015,26(17):2344-2351+2373.

[3] Hu Y, Yin M, Li X. A novel objective function for job-shop scheduling problem with fuzzy processing time and fuzzy due date using differential evolution algorithm[J]. The International Journal of Advanced Manufacturing Technology, 2011, 56(9): 1125-1138.

[4] Gu Feng, Chen Hua-ping, Lu Bing-yuan. Optimization for fuzzy flexible jobshop scheduling based on genetic algorithm[J].Systems Engineering and Electronics, 2006(07):1017-1019+1038.

[5] Deming Lei. Population-based neighborhood search for job shop scheduling with interval processing time[J]. Computers & Industrial Engineering, 2011, 61(4) : 1200-1208.

[6] Matsveichuk N M, Sotskov Y N, Egorova N G, et al. Schedule execution for two-machine flow-shop with interval processing times[J]. Mathematical and Computer Modelling,2009, 49(5-6): 991-1011.

[7] Yang hong-an, Wang Zhou-feng, Lv Yang-yang. Interval number solving method for job-shop scheduling problem with processing time variability[J].Computer Integrated Manufacturing Systems, 2014,20(09):2231-2240.

[8] Peng Jiangang, Liu Mingzhou, Zhang Xi, Zhang Mingxin, Ge Maogen. Rescheduling Algorithm Based on Uncertain Processing Time of Operations for a Flexible Job-shop[J].China Mechanical Engineering, 2014,25(17):2320-2326.

[9] Chen Yuxuan. Solution about the Interval Number of Flexible Job Shop Scheduling Problem Based on Improved Genetic Algorithm under Uncertain Time[J]. Mechanical Engineer, 2018(1):74-76.

[10] Jia Chunfu. Single machine stochastic scheduling with Exponential distribution processing time[J]. Systems Engineering, 2002(06):58-61.

[11] Manna D K. Common due-date assignment and scheduling on single machine with exponential processing times[J]. Opsearch, 2000, 37(3): 221-236.

[12] Jia C. Stochastic single machine scheduling with an exponentially distributed due date[J]. Operations Research Letters, 2001, 28(5): 199-203.

[13] Wang kanghong. Simulation and optimization of job shop scheduling with place buffer under uncertain disturbances[J].Journal of Light Industry, Vol.35 No.6 Nov.2020.

[14] Jamal Shahrabi and Mohammad Amin Adibi and Masoud Mahootchi. A reinforcement learning approach to parameter estimation in dynamic job shop scheduling[J]. Computers & Industrial Engineering, 2017, 110 : 75-82.

[15] Sobaszek Ł, Gola A, Kozłowski E. Job-shop scheduling with machine breakdown prediction under completion time constraint[C]//2018 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, 2018: 437-440.

[16] Tang qiuhua, Chen Shijie, Zhao Meng, Zhang Liping. Prediction of Optimal Rescheduling Mode under Machine Failures within Job Shops[J].China Mechanical Engineering,2019,30(02):188-195.

[17] Ba Zhiyong, Yuan Yiping, Pei Guoqing2, Wang Bo. Hybrid evolutionary algorithm with multi-operation precise joint movement neighborhood structure for the job shop scheduling problem[J]. Computer Integrated Manufacturing Systems:1-29[2023-07-08].http://kns.cnki.net/kcms/detail/ 11.5946.tp. 20230104.1157.008.html.

[18] Liu Xuan, Shang Jun, Bai Ao. Production task rescheduling method for job shop based on equipment failure[J].Manufacturing Automation, 2016,38(12):26-30+60.