Robotics and Computer–Integrated Manufacturing 71 (2021) 102141

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Research on intelligent workshop resource scheduling method based on improved NSGA-II algorithm

Minghai Yuan *, Yadong Li, Lizhi Zhang, Fengque Pei

College of Mechanical and Electrical Engineering, Hohai University, Changzhou, China

ARTICLE INFO

Keywords:
Intelligent manufacturing
Workshop resource scheduling
Improved NSGA-II algorithm
Variable proportion

ABSTRACT

With the intensification of globalization, the competition among various manufacturing enterprises has become increasingly fierce, enterprises are developing in the direction of the product diversification, zero inventory or low inventory, and scheduling in production management has become more complicated. In this paper, machine and workpiece were as objects to study the problem of workshop scheduling in intelligent manufacturing environment. The resource scheduling model of intelligent manufacturing workshop was established with the goal of minimizing the maximum completion time, tardiness, machine load and energy consumption. The Non-dominated Sorting Genetic Algorithm (NSGA-II) algorithm was improved, and the evaluation function was established based on ranking level and crowding degree, then the competition mechanism was introduced. Random mutation strategy and crossover method based on process and machine was adopted to generate a new generation of populations. The elitist retention strategy was improved, the variable proportion method was designed to determine the probability, and the optimal solution is determined by the Analytic Hierarchy Process (AHP). The benchmark cases and practical production and processing problems were tested to verify the superiority and effectiveness of the improved algorithm.

1. Introduction

Management has become an important factor in the development of modern enterprises, as an important aspect of management, production scheduling is one of the restrictive factors in the production process of enterprises [1]. The international society of production engineering has found that every advanced manufacturing mode is based on scheduling, and production scheduling has become the core of modern manufacturing and management [2]. Production scheduling is a resource allocation problem, which allocates limited resources to several tasks to meet one or multiple optimization goal, the essence is to realize the mapping of unit processing task decomposed by production task to corresponding processing resource [3–5]. In intelligent workshop environment, the multi-source job shop data can be obtained in real time, big data can be analyzed in real time, the error between the set processing time and the actual processing time can be eliminated, and the cloud-edge computing scheduling system can be adopted to ensure fast and accurate solution of the algorithm.

There are precise algorithms and approximate algorithms for job shop scheduling. The accurate algorithms can ensure the global optimal solution, but it is only suitable for small-scale and simple environment problem. When the scale of the problem is large, especially in the digital and integrated intelligent manufacturing environment, the problem is difficult to be solved by the complicated accurate solving algorithms [6]. The approximation methods include heuristic rules and intelligent optimization algorithms. Heuristic rules include rules based on priority assignment, methods based on insertion and rules based on shifting bottleneck. The priority assignment rule is to arrange the processing order of the workpiece according to the set priority. There are rules such as first-come first-serviced, maximum number of remaining processes first, earliest deadline first, shortest processing time first, longest processing time first, etc. The implementation process of rules based on insertion is complicated, and the scheduling process needs to be continuously sorted according to the workpiece status and machine status, including CDS rules, Johnson rules, NEH rules, etc. [7]. Heuristic rules cannot quantitatively evaluate the performance of the solution, and it is difficult to guarantee the optimality and feasibility of the obtained solution. Intelligent optimization algorithm has the characteristics of weak problem dependence and does not require a deep analysis of specific issues. The entire search process can be completed only through

* Corresponding author.
E-mail addresses: ymhai@hhu.edu.cn (M. Yuan), 2425492198@qq.com (Y. Li), 1303276961@qq.com (L. Zhang), 20201013@hhu.edu.cn (F. Pei).

https://doi.org/10.1016/j.rcim.2021.102141
Received 31 July 2020; Received in revised form 18 February 2021; Accepted 18 February 2021
Available online 2 March 2021
0736-5845/

computer iteration, and the satisfactory solution can be obtained in a short time [8,9].

D. Bai [10] designed a hybrid discrete differential evolution algorithm to schedule job shop resources with the goal of minimizing the total weighted completion time. T. Jiang [11] established the job shop scheduling model with the objective of minimizing the sum of the energy consumption cost and the earliness/tardiness cost, and adopted the bi-population optimization algorithm based on discrete cat swarm algorithm to solve the model, two sub-populations were used to adjust the machine operation and allocation order respectively, six adjustment curves were used to change the number of cats in the seeking mode and tracking mode to coordinate the global search and local search of each sub-population, and the information exchange strategy was introduced to realize the cooperation between the two sub-populations. L. Yin [12] regarded the machining spindle speed as an independent decision variable to establish a low-carbon scheduling mathematical model, and used a simplex lattice-based multi-objective genetic algorithm to solve it. Y. Laili [13] studied multi-phase integrated scheduling of hybrid tasks in cloud manufacturing environment, five key objectives were considered and six representative multi-objective evolutionary algorithms were compared and adopted to solve the problem, C. Li [14] analyzed the energy consumption in the job shop production, he used the simulated annealing algorithm to solve the optimization model with the goal of minimizing the completion time and energy consumption in the job shop, and realized the flexible batch optimization scheduling of multiple process routes. In addition, he also studied the processing energy consumption under dynamic events, and proposed a gravity search algorithm to solve the rescheduling model. S. Zhao [15] aimed at minimizing the completion time and designed a hybrid algorithm with improved neighborhood structure. The concept of virtual process and virtual working hour is introduced by C. Wang [16] to establish the scheduling model of toy factory, combined with the strategy of dynamic event scheduling and cycle scheduling, the dynamic scheduling is transformed into multiple continuous static scheduling and solved by genetic algorithm, the effectiveness of the proposed algorithm is verified in the mold shop. Tabu search algorithm has strong global search ability, and particle swarm optimization has strong local search ability, T. Mao [17] combined particle swarm optimization algorithm with tabu search algorithm, he used three kinds of strategy moving operators and proposed a dynamic judgment mechanism of population stagnation, and designed a hybrid particle swarm optimization algorithm. M. Dai [18] proposed an enhanced genetic algorithm and designed a multi-objective optimization model with the objective of minimizing energy consumption and makespan, and adopted the improved genetic algorithm to solve it. X. Liang [19] proposed a parallel and adaptable immune algorithm, which used adaptive crossover operator and mutation operator to perform immune operations on antibodies. X. Huang [20] used the improved ant colony algorithm to study the flexible job-shop scheduling problem with multiple process plans, the optimization goal was to minimize the maximum completion time, ant path was determined based on OR subgraph, Allowed list and Tabu list, simulation results verified the effectiveness of the proposed algorithm.M. Li [21] proposed an empire competition algorithm with diversity operators to solve flexible job shop scheduling problem with the objectives of processing time, total delay, total load and total energy consumption. L. Feng [22] designed a two-level multi-task scheduling model, two scheduling strategies were presented and evaluated, the two-level two-optimization scheme in particular can find a better schedule. G. Zhang [23] considered machine failures in job shop scheduling and proposed robustness indicators, machine failure events were represented by established fault occurrence functions, and genetic algorithm was used to optimize the indicators. In addition, artificial bee colony algorithm [24], gray wolf optimization algorithm [25], fruit fly optimization [26], whale colony algorithm [27], water wave optimization algorithm [28] are also used to solve the job shop scheduling problem.

Job shop scheduling is a non-deterministic polynomial-time hard (NP-hard) problem. There are many methods for job shop scheduling, however, these algorithms have some shortcomings in solving accuracy and stability. Many algorithms are easy to fall into local optimization or have the problem of large amount of computation. Genetic algorithm has powerful optimization capabilities, especially in the multi-objective, large-scale, high-dimensional intelligent job shop scheduling environment. Genetic algorithm uses evolution process to obtain information and organize the search by itself. Individuals with greater fitness have higher survival probability and can obtain the gene structure more suitable for environment. This paper used the characteristics of self-organization, self-adaptation, and self-learning of genetic algorithm and drew on the idea of Pareto optimal solution set to improve the Non-dominated Sorting Genetic Algorithm (NSGA-II) algorithm to solve the problem of job shop scheduling.

Table 1
Symbol description.

Parameter symbol	description
J_i	The total number of processes for workpiece N_i
CT_{ie}	Processing end time of workpiece N_i
$CTijs$	Processing start time of O_{ij}
$CTije$	Processing end time of O_{ij}
DT_{ie}	Delivery date of workpiece N_i
$ctijk$	Time for machine M_k to process O_{ij}
$xijk$	0–1 variable, which indicates whether O_{ij} is processed on the *k-th* machine tool, if it is 1, then it is, otherwise it is 0
$P1$	Fixed power of workshop
$W2$	Energy consumption of workpiece transfer once
$dcon$	Total number of workpiece transfers
pmr	No-load power of machine tool
pmw	Machining power of machine tool
Tmr	No-load time of machine tool
Tmw	Processing time of machine tool

In this paper, the mathematical model for processing N workpieces on M machines was established by considering time factors, machine load factors, and energy consumption factors:.

2. Multi-objective single-resource scheduling optimization model

2.1. Analysis of job shop scheduling problem

The job shop scheduling problem in intelligent manufacturing environment can be described as: there are N workpieces to be processed and M machines available for processing in a processing system. Each workpiece N_i $(i = 1,2,\ldots,n)$ contains one or more operations, O_{ij} represents the *j-th* operation of the workpiece N_i, and each operation O_{ij} can be processed by any one of the optional machines set $M_{ij} \subseteq \{M_1, M_2, \cdots, M_m\}$. The processing time of the operation changes with the change of the processing machine, and the operation sequence of the workpiece has been determined in advance. In order to study the workshop scheduling problem in intelligent manufacturing environment, the following constraint assumptions need to be met::

1) The same machine can only process one workpiece at the same time;
2) The same workpiece can only be processed by one machine at the same time;
3) Each machine and each workpiece has the same priority. Once the workpiece starts to be processed, it can not be interrupted unless the machine breaks down;
4) The moving time of each process between the processing machines is 0;
5) The time when the machine starts to process the workpiece is the start-up time, and the time when the machine finishes the last operation is the shutdown time.

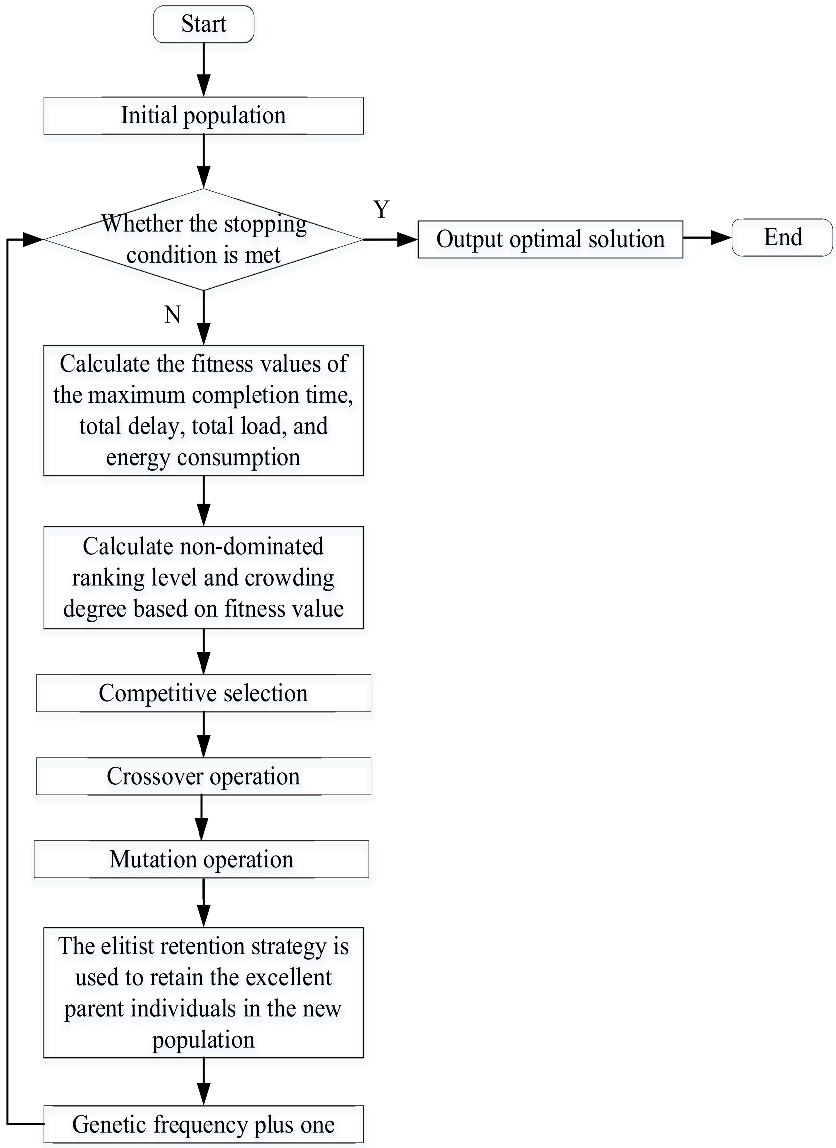

Fig. 1. General flow of improved NSGA-II algorithm.

The description of relevant symbols involved in the subsequent scheduling model is shown in Table 1.

$$F_{goal} = \min[f_1, f_2, f_3, f_4] \tag{1}$$

$$f_1(x) = \max\{CT_{ie}\} \tag{2}$$

$$f_2(x) = \sum_{i=1}^{N} \max(CT_{ie} - DT_{ie}, 0) \tag{3}$$

$$f_3(x) = \sum_{i=1}^{N} \sum_{j_i=1}^{J_i} \sum_{k=1}^{M} ct_{ijk} x_{ijk} \tag{4}$$

$$f_4(x) = p_1 \cdot \max\{CT_{ie}\} + w_2 \cdot d_{con} + \sum_{m=1}^{M} (p_{mr} \cdot T_{mr} + P_{mw} \cdot T_{mw}) \tag{5}$$

$$CT_{ije} = CT_{ijs} + ct_{ijk} \tag{6}$$

$$CT_{i(j+1)s} \geq CT_{ije} \tag{7}$$

$$\sum_{k=1}^{M} x_{ijk} = 1 \tag{8}$$

Eq. (1) shows that the minimum value of the four objective functions is the optimal solution, f_1 is the maximum completion time, f_2 is the total delay of the machine tool, f_3 is the total load of the machine tool, f_4 is the processing energy consumption, Eqs. (6) and (7) indicate that the same workpiece needs to be processed in a certain order, and each machine can only process one workpiece at the same time, Eq. (8) shows that the same workpiece can only be processed by one machine at the same time.

2.2. . Job-shop scheduling algorithm based on improved NSGA-II

Job shop scheduling in intelligent manufacturing environment has the characteristics of multi-objective, high-dimensional and large-scale, which is the NP-hard problem [29]. Genetic algorithm is an intelligent optimization algorithm based on natural evolution and selection mechanism. It has strong global optimization ability. The evaluation function is used in the search process, and the solution process is simple. Probability mechanism is used to iterate at random, which has been

Process											Machine										Objective function			
7	8	6	2	3	2	2	1	4	5	...	3	1	2	4	7	5	4	4	6	4	...	88	12	178	160

Fig. 2. Chromosome encoding diagram.

widely used in production scheduling, machine learning, combination optimization, image signal processing, adaptive control and artificial life [30,31]. However, the general genetic algorithm has the shortcomings of being easy to fall into premature and slower search speed in the later stage. In this paper, the improved NSGA-II algorithm is used to solve the intelligent manufacturing workshop scheduling problem.

The overall flowchart of the improved NSGA-II algorithm is shown in Fig. 1, firstly, the initial population was randomly generated based on the workpiece operation and the corresponding processing machine, and the greedy decoding algorithm was designed to calculate the maximum completion time, total delay, total load and energy consumption of each chromosome, and each chromosome was evaluated according to the ranking level and crowding degree. The competitive bidding method was used to select the chromosome for genetic operation. The new population was generated by crossover operation and mutation operation based on workpiece process and machine. The elitist retention strategy was used to select the excellent parent individuals in the new population, and the optimal solution was output after the iteration.

The basic operation of the algorithm is as follows:

(1) Population initialization

In this paper, a multi-layer coding method based on real number is adopted. and each chromosome represents a feasible solution of the problem to be optimized. The population size is NIND, the total number of operations for all workpieces to be processed is *P_number*, the number of objective functions is *G_number*, the length of the initial chromosome equals twice *P_number* plus *G_number*. The first *P_number* genes represent the workpiece operation process, the following *P_number* genes represent the corresponding processing machine, and the last G_number genes represent the objective function value. The first *P_number* of the chromosome workpiece operations is generated randomly, and the second *P_number* of the processing machines is randomly selected from the set of optional machines. As shown in Fig. 2, the 7 on the first gene locus represents the process O_{71}, which is processed by the machine M_3. The 2 on the gene locus 4, 6 and 7 respectively represent process O_{21}, O_{22} and O_{23}, and are respectively processed by the machines M_4, M_5 and M_4. The objective function gene from left to right represent the maximum completion time, total delay, total machine load and total energy consumption.

(2) Determination of the objective function fitness value

The maximum completion time is the time when all workpieces are processed. The greedy decoding algorithm is introduced into decoding calculation in this paper. For a certain workpiece, the completion time of the previous operation of the workpiece is *end_t_p*, and the completion time of the last operation of the machine processing is *end_t_m*. Compare *end_t_p* and *end_t_m*, if *end_t_p* ⟨ *end_t_m*, then find out the machine processing gap that meets the workpiece forward insertion. The time of machine gap is *span_time*, the workpiece processing time is *pro_time*, the time of machine processing gap needs to meet two conditions at the same time, the first is *span_time*⟩ *pro_time*, the second is the last moment of the machine gap time is bigger than *end_t_p* plus *pro_time.* If there are multiple optional machining gaps, the one with the earliest completion time is preferred. If there is no qualified machine gap, then the start machining time of the workpiece is *end_t_m*. If *end_t_p*≥*end_t_m*, the workpiece start processing time is *end_t_p*. The total delay is the sum of the time that all workpieces exceed the specified delivery time. The total load is the sum of all machine processing time. The total energy consumption is the sum of the job shop inherent energy consumption, the machine no-load energy consumption, the machine working energy consumption and the workpiece transfer energy consumption.

(3) Fast non-dominated and crowding ranking

Firstly, the Pareto grade of each individual is determined according to the number of each individual dominated in a population and the set of individuals dominated by the individual. In order to ensure the population diversity and make the population evolve towards a better direction, the crowding degree is introduced into the algorithm, which represents the density of other individuals around an certain individual in the same non-dominated level.

Firstly, the objective function values in the first column are sorted in ascending order, and the individual distance between the maximum objective function value and minimum value is set to infinity, and the distance between the remaining individuals is calculated according to $\frac{next\_obj - previous\_obj}{f\_max - f\_min}$. *next_obj* represents the next objective function value of this individual, and *previous_obj* represents the previous objective function value of this individual. *f*_max and *f*_min represent the maximum and minimum values of the objective function in this column. The other column objective function values are treated in the same way, and finally the crowding degree is obtained by adding the four objective distance values of each individual.

(4) Competition selection

Two individuals are randomly selected from the population for genetic operation, and the individuals with higher ranking level are preferred (in this algorithm, the ranking level 1 is the highest). If the ranking level is the same, the individual with large crowding degree is preferred; if the crowding degree is the same, one of the individuals is selected randomly.

(5) Crossover and mutation operation

In the early evolution, the fitness of chromosomes is poor and requires a large probability of crossover and mutation. In the later evolution, the chromosome already has a good structure and requires a small probability of crossover and mutation. Therefore, in this paper, variable probability crossover and mutation is adopted, that is, the probability of crossover and mutation decreases linearly from the first generation to the *N-th* generation. In order to show the operation process more clearly, some chromosomes of 8 × 8 (8 workpieces and 8 machines) are selected below to illustrate the crossover and mutation method, in which the objective function gene is not listed.

1) Crossover operation

In this paper, the hybrid crossover mode based on workpiece operation process crossover and machine crossover is adopted. The *rand* function is used to generate random number $r (r \in [0, 1])$. If $r \leq 0.5$, process-based crossover is adopted, otherwise machine-based crossover is used. The specific operation is shown in the figure below.

For the process-based crossover, the workpieces are divided into two groups, and the function *randi([0,1],[1,N])* is used to randomly generate a 0,1 matrix: $R_{1\times N}$ (N is the number of the workpieces). The workpieces with the position 1 in the matrix are a group, and the rest are a group. As

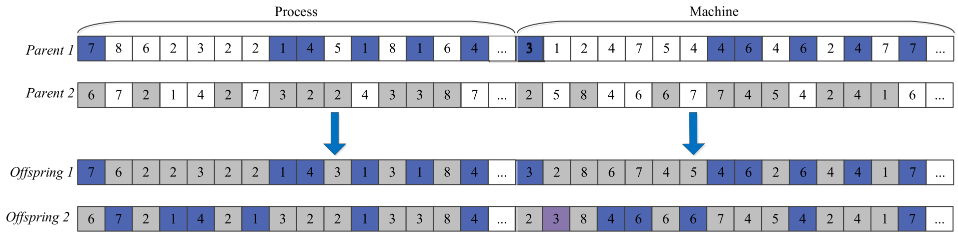

Fig. 3. Process-based crossover.

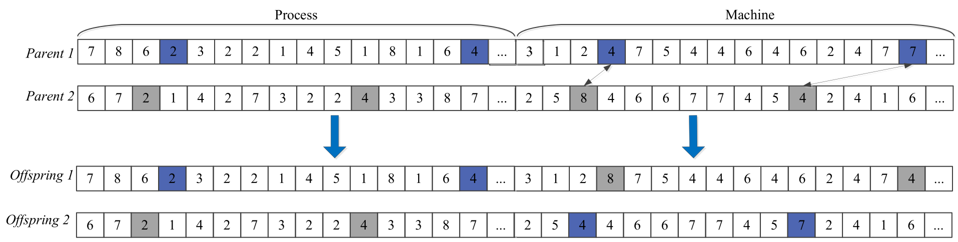

Fig. 4. Machine-based crossover.

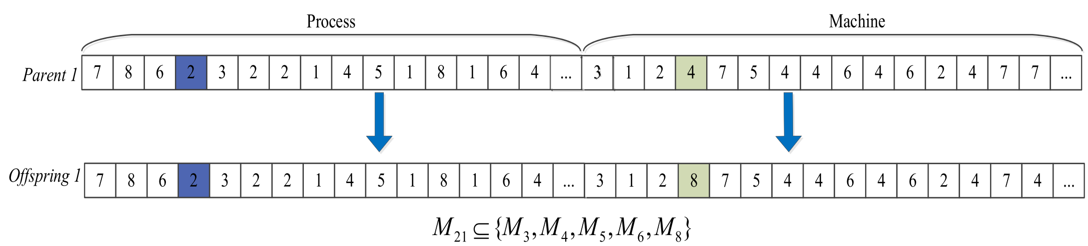

Fig. 5. Mutation operation.

shown in Fig. 3, workpieces 1, 4, and 7 are a group, and workpieces 2, 3, 5, 6, and 8 are a group. The gene of workpiece 1, 4, 7 and the corresponding machine gene in the parent chromosome 1 remains unchanged, the gene of workpiece 2, 3, 5, 6, 8 and the corresponding machine gene in parent chromosome 2 is successively inserted into the remaining gene locus of the parent chromosome 1. In the same way, the gene of workpiece 2, 3, 5, 6, 8 and the corresponding machine genes in the parent chromosome 2 remains unchanged, the gene of workpiece 1, 4, 7 and the corresponding machine gene of parent chromosome 1 is successively inserted into the remaining gene locus of the parent chromosome 2.

For the machine-based crossover, a 0,1 matrix with the size equal to the length of the workpiece process chromosome is randomly generated, and the processing machine corresponding to the number 1 on the parent chromosome 1 is crossed. Assuming that the gene position corresponding to the number 1 is 4,15…, and taking the gene position 15 as an example. As shown in Fig. 4, the gene locus 15 of the parent chromosome 1 is O_{42}, which is processed by M_7, and O_{42} on parent chromosome 2 is processed by M_4. The processing machines of the O_{42} in the two parent chromosomes are crossed, that is, the O_{42} in parent chromosome 1 is processed by M_4, and the O_{42} in parent chromosome 2 is processed by M_7.

1) Mutation operation

The *randperm(P_number,cal)* function is used to randomly generate chromosome mutation positions, *cal* represents the number of chromosome gene mutations. Assuming that one of the mutation positions is the 4-th gene. As shown in Fig. 5, the processing machine of O_{21} is mutated, and the optional machine set of O_{21} is M_3, M_4, M_5, M_6, M_8. The processing machine for O_{21} in the parent chromosome 1 is M_4, and another machine in the machine set is randomly selected for processing during mutation. For example, machine M_8 is selected.

(6) Elitism preserving strategy

The recessive elitism preserving strategy is adopted in traditional NSGA-II. Firstly, the parent and offspring are merged to form the total population Pop_t. Secondly, the individuals with high rank and high crowding degree in the population Pop_t are selected according to the fast non-dominated and crowding ranking. This method can effectively improve the convergence of genetic algorithm. However, the new individuals generated in the population will be rejected in the later stage, which is not conducive to the diversity of the population and easy to make the algorithm fall into local optimization. In this paper, the variable ratio method is designed, the traditional elitism preserving strategy is used in the previous *N/3* generations, and the variable proportion method is designed from the *N/3* generation to the *N-th* generation. the variable ratio method is designed to make the proportion of the optimal parent generation in the population decreases linearly with the increase of iteration number.

(7) Determination of the optimal solution

When the algorithm iteration stop condition is satisfied, the optimal solution set is output. For the multi-objective job shop scheduling problem, there are conflicts among objective functions, and the satisfactory optimal solution can be selected from the optimal solution set

Table 2
The scale determination of positive reciprocal matrix.

Scale	Meaning
1	Factor *i* is as important as factor *j*
3	Factor *i* is slightly more important than factor *j*
5	Factor *i* is significantly more important than factor *j*
7	Factor *i* is strongly more important than factor *j*
9	Factor *i* is extremely more important than factor *j*

Note: 2, 4, 6 and 8 are the intermediate values of the comparison between factor *i* and factor *j*

Table 3
The Values of random consistency index *RI*.

m	1	2	3	4	5	6	7	8	9
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

according to the ranking level and crowding degree. In order to better reflect the importance of some indicators for the enterprise, The Analytic Hierarchy Process (AHP) is adopted in this paper to determine the weight of each index and select the optimal solution.

AHP is a multi-objective decision-making method for dealing with limited schemes [32], which was proposed by T.L.Saaty of University of Pittsburgh in 1970s. The method has the advantages of less information requirement and short decision-making time. The steps of analytic hierarchy process are as follows:

1) Construct a positive reciprocal matrix

$A = (a_{ij})_{m\times m} = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1m} \\ a_{21} & 1 & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & 1 \end{bmatrix}$, Where $a_{ji} = \frac{1}{a_{ij}}$, and its value is determined by Table 2. For example, The maximum completion time is compared with the total equipment load. If the former is significantly more important than the latter, then a_{12}=5, a_{21}=1/5. Each indicator is compared with other indicators, and finally the reciprocal matrix A is determined

2) Consistency check

a) Normalize each column vector of A matrix$\widetilde{w}_{ij} = a_{ij} / \sum_{i=1}^{m} a_{ij}$;
b) Sum$\widetilde{w}_{ij}$in rows to get$\widetilde{w}_i = \sum_{j=1}^{m} \widetilde{w}_{ij}$;
c) Normalize$\widetilde{w}_i$to$w_i = \widetilde{w}_i / \sum_{i=1}^{m} \widetilde{w}_i$, $w = (w_1, w_2, \cdots, w_m)^T$ is the eigenvector;
d) Calculate $\lambda = \frac{1}{m} \sum_{i=1}^{m} \frac{(Aw)_i}{w_i}$.

The consistency index $CI = \frac{\lambda - m}{m-1}$, m is the order of the positive reciprocal matrix A. The CI is compared with the same order random consistency index RI in Table 3, that is, the consistency ratio is $CR = \frac{CI}{RI}$. When $CR<0.1$, it means that the positive reciprocal matrix A has passed the consistency check, otherwise the positive reciprocal matrix A needs to be properly modified. When the positive reciprocal matrix A passes the consistency test, the eigenvector $w = (w_1, w_2, \cdots, w_m)^T$ is the required weight vector.

3) Comprehensive evaluation

After the weight of each objective function is determined, then the optimal solution set data is normalized according to Eq. (9) to obtain matrix B, where$B = (target_p_{ij})_{q\times 4}$, $target\_p_{ij}$ represents the data of the *j-th* column and the *i th* row of the optimal solution set after normalization. The scheduling scheme corresponding to the minimum value min(C)in the comprehensive evaluation matrix $C = Bw = \sum_{j=1}^{4} target\_p_{ij} w_j$ is the optimal scheme.

Table 4
Matrix relationship table for comparison of various factors.

	maximum completion time	total delay	total equipment load	energy consumption
maximum completion time	1	3	5	7
total delay	1/3	1	3	4
total equipment load	1/5	1/3	1	2
energy consumption	1/7	1/5	1/2	1

Table 5
Comparison of results of 8 × 8 test cases.

objective function	SPT	GA	AL	AL+CGA	PSO+SA	NSGA-II	
maximum completion time f_1	19	16	16	16	16	16	15
total machine load f_3	91	77	75	75	73	73	73

$$target\_p_{ij} = \begin{cases} \dfrac{target_{ij} - target_j^{min}}{target_j^{max} - target_j^{min}} & \left(target_j^{max} \neq target_j^{min}\right) \\ 0 & \left(target_j^{max} = target_j^{min}\right) \end{cases} \tag{9}$$

$$target_j^{max} = \max\left(target_{1j}, target_{2j}, \cdots, target_{qj}\right) \tag{10}$$

$$target_j^{min} = \min\left(target_{1j}, target_{2j}, \cdots, target_{qj}\right) \tag{11}$$

3. Simulation and analysis

In order to verify the effectiveness of the improved NSGA-II algorithm proposed in this paper, The benchmark cases and practical problems are used to verify the superiority and effectiveness of the algorithm. Matlab 2016b is used to solve the multi-objective NSGA-II algorithm in this paper. The hardware platform is *Intel(R) Core(TM) i5–8400 CPU @2.80GHz, RAM 16 GB*. The basic parameters of the improved NSGA-II algorithm proposed in this paper are set as follows: the initial population size NIND is 200, the maximum iteration number GEN is 50, the crossover probability P_c is 0.8→0.4, and the mutation probability Pv is 0.1→0.02, the elitism preserving strategy number of parent/offspring 40% →10%, The energy consumption of workpiece transfer once W_2 is 0.3, The fixed power of workshop P_2 is 35. The matrix relationship of various factors is show in Table 4.

According to the matrix relationship Table 4, the weight of the maximum completion time, total delay, total equipment load, and energy consumption is w=(0.5694,0.2546,0.1100,0.066)^T, and CR=0.0217<0.1, which meets the consistency requirements.

3.1. . Benchmark test

In this paper, the benchmark problem in reference [33] is used to test. The example is an 8 × 8 partially flexible scheduling problem. Table 5 shows the comparison results of the Shortest Processing Time (SPT), Genetic Algorithm (GA), genetic algorithm controlled by the assigned model which is generated by the approach of localization (AL), Controlled Genetic Algorithm with Approach by Localization (AL+CGA), Particle Swarm Optimization based Simulated Annealing Algorithm (PSO+SA) algorithm and the improved NSGA-II algorithm proposed in this paper(The maximum completion time f_1 and total machine load f_3 of the objective function in reference [35] are the same as those in this paper, so only the two objective functions are compared in this paper). For the improved NSGA-II algorithm, the value in the first

Table 6
Case data of intelligent manufacturing job-shop scheduling.

Workpiece N_i	release time	Process O_{ij}	processing time/min										delivery date/min
			M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	
J_1	0	O_{11}	–	–	18	10	16	–	18	12	–	–	80
		O_{12}	15	12	–	–	–	18	–	17	16	–	
		O_{13}	17	–	12	–	9	–	11	–	–	10	
J_2	0	O_{21}	–	14	–	17	16	13	–	15	–	–	100
		O_{22}	18	–	24	–	16	–	20	–	22	–	
		O_{23}	13	16	–	15	–	16	17	–	–	16	
		O_{24}	–	7	–	6	–	10	–	8	8	–	
J_3	0	O_{31}	16	–	18	–	14	–	20	–	19	–	–
		O_{32}	11	10	–	16	–	8	–	12	–	–	
J_4	0	O_{41}	21	–	21	–	20	–	18	–	–	17	100
		O_{42}	–	24	–	20	–	18	–	22	21	–	
		O_{43}	6	7	–	5	7	6	–	9	–	10	
J_5	0	O_{51}	–	–	12	–	15	–	–	14	–	16	–
		O_{52}	15	–	–	12	–	–	13	–	–	–	
		O_{53}	–	10	11	–	15	–	16	–	15	–	
J_6	0	O_{61}	–	15	–	17	–	16	14	–	–	19	50
		O_{62}	14	–	12	–	15	–	–	14	–	13	
J_7	0	O_{71}	18	–	15	–	–	24	–	–	20	–	–
		O_{72}	–	14	–	17	–	20	15	12	–	15	
		O_{73}	8	10	9	–	7	12	–	7	8	9	
		O_{74}	15	20	18	23	–	16	–	16	–	–	
		O_{75}	–	25	–	–	15	–	18	17	20	–	
		O_{76}	10	8	7	7	7	8	9	–	6	8	
J_8	0	O_{81}	17	20	–	23	19	19	–	15	16	–	120
		O_{82}	–	18	15	20	–	22	19	17	–	–	
		O_{83}	24	–	26	24	25	24	–	–	–	20	
		O_{84}	–	22	–	18	–	–	19	–	20	–	
J_9	0	O_{91}	5	7	7	11	8	11	10	–	8	–	80
		O_{92}	24	25	–	22	–	18	–	20	–	21	
		O_{93}	15	–	13	13	14	10	–	11	–	–	
J_{10}	0	$O_{10,1}$	20	–	18	–	19	18	17	–	22	–	100
		$O_{10,2}$	15	14	–	15	–	16	13	15	–	12	
		$O_{10,3}$	11	–	8	12	10	13	–	8	9	9	

Table 7
Machine tool processing power and no-load power.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
processing power p_{mw}/Kw	20	15	8	12	7	16	14	18	6.5	10
no-load power p_{mr}/Kw	3	2	0.5	0.64	0.45	1.4	1.1	1.5	0.35	0.8

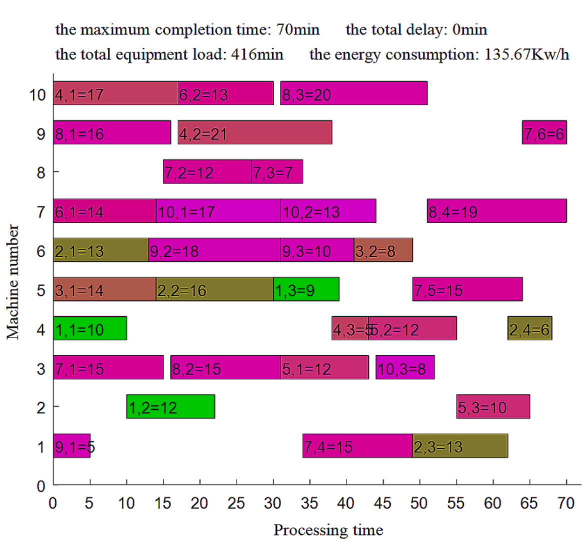

Fig. 6. 10×10 example simulation processing Gantt chart.

column are the results obtained in multi-objective operation environment, and the value in the latter column are the results obtained in single-objective operation environment. It can be seen from Table 5 that the optimal values obtained by the improved NAGA-II algorithm are better than or equal to the best values of other algorithms, the improved NAGA-II algorithm can get good results for intelligent manufacturing shop scheduling, which verifies the effectiveness and superiority of the proposed algorithm.

3.2. . Simulation

According to the actual data of the intelligent manufacturing job shop, Table 6 shows the release time, workpiece operation process, delivery date and processing time of 10 workpieces. The number in the processing time column indicates the processing time, the symbol "-" indicates that the workpiece operation cannot be processed by the machine M_k. Table 6 and the following enterprise processing time is the precise setting time after making up the difference. The number in the delivery date column indicates the delivery date required by the customer, the symbol "-" means that the workpiece needs to be processed as soon as possible, and there is no time requirement. Table 7 shows the processing power and no-load power of the processing machine.

The final gantt chart of scheduling scheme is shown in Fig. 6.

It can be seen from the figure that the maximum completion time is 70 min, the total delay is 0 min, the total equipment load is 416 min, and the energy consumption is 135.67Kw/h. 4,1 = 17 means that the processing time of process O_{41} is 17 min in the Fig. 6.

4. Conclusion

This paper studied the problem of workshop resource scheduling in the intelligent manufacturing environment. With the goal of minimize the maximum completion time, total tardiness, total load, and total energy consumption, the resource scheduling model for intelligent manufacturing workshops was established. The NSGA-II algorithm was improved, and an evaluation function based on ranking levels and congestion was established. The competition mechanism was introduced to select the excellent individuals, and the crossover and random mutation methods based on process and equipment were used to generate a new generation of population. The elitism preserving strategy was improved and the AHP was used to determine the optimal solution. The superiority and effectiveness of the improved algorithm were verified by testing benchmark cases and actual processing problems. In the future, we will continue to study on improving the convergence speed and stability of the algorithm.

Author statement

Minghai Yuan, Yadong Li, Lizhi Zhang, Fengque Pei acknowledge that the material presented in this manuscript has not been previously published, except in abstract form, nor is it simultaneously under consideration by any other journal.

The overarching research goals were developed by Minghai Yuan and Fengque Pei. Minghai Yuan and Yadong Li carried out conceptualization, methodology, software, visualization, validation, writing-original draft. Lizhi Zhang and Fengque Pei conducted investigation, literature review.

Declaration of Competing Interest

We have no conflicts of interest to disclose.

Acknowledgment

This work was supported by General program of Natural Science Foundation of Jiangsu Province under Grant number BK20201162, the Changzhou Science and Technology Support Plan (Social Development) under Grant number CE20205045, the National Nature Science Foundation of China under Grant number 51875171.

Reference

[1] X. Yao, H. Jin, J. Zhang, Towards a wisdom manufacturing vision, Int. J. Comput. Integr. Manuf. 28 (12) (2015) 1291–1312.
[2] S. Wang, M. Liu, C. Chu, A branch-and-bound algorithm for two-stage no-wait hybrid flow-shop scheduling, Int. J. Prod. Res. 53 (4) (2015) 1143–1167.
[3] M. Yuan, Z. Zhou, X. Cai, C. Sun, W. Gu, Service composition model and method in cloud manufacturing, Robot. Comput. Integr. Manuf. 61 (2020).
[4] S. Cheng, W. Ji, Y. Chou, K. Jiang, Improved bird swarm algorithm for dual resource constrained discrete intelligent job shop scheduling problem, Mod. Manuf. Eng. 04 (2019) 20–26.
[5] M. Cheng, H. Zhu, Z. Zhang, Y. Jin, Y. Wang, D. Tang, Multi-agent job shop scheduling strategy based on pheromone, China Mech. Eng. 29 (22) (2018) 2659–2665.
[6] M. Torkashvand, B. Naderi, S. Hosseini, Modelling and scheduling multi-objective flow shop problems with interfering jobs, Appl. Soft Comput. 54 (2017) 221–228.
[7] D. Zhang, X. Li, Fast heuristic algorithm for job shop scheduling problem, Comput. Integr. Manuf. Syst. 02 (2005) 237–241.
[8] K. Baizid, A. Yousnadj, A. Meddahi, R. Chellali, J. Iqbal, Time scheduling and optimization of industrial robotized tasks based on genetic algorithms, Robot. Comput.-Integr. Manuf. 34 (2015) 140–150.
[9] J. Zhang, W. Wang, X. Xu, A hybrid discrete particle swarm optimization for dual-resource constrained job shop scheduling with resource flexibility, J. Intell. Manuf. 28 (8) (2017) 1961–1972.
[10] D. Bai, Z. Zhang, Q. Zhang, Open shop scheduling problem to minimize total weighted completion time, Eng. Optim. 49 (1) (2017) 98–112.
[11] T. Jiang, G. Deng, Optimizing the low-carbon flexible job shop scheduling problem considering energy consumption, IEEE Access 6 (2018) 46346–46355.
[12] L. Yin, X. Li, L. Gao, C. Lu, A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem, Sustain. Comput.-Inf. Syst. 13 (2017) 15–30.
[13] Y. Laili, S. Lin, D. Tang, Multi-phase integrated scheduling of hybrid tasks in cloud manufacturing environment, Robot. Comput. Integr. Manuf. (2020) 61.
[14] C. Li, H. Shen, L. Li, A batch splitting flexible job shop scheduling model for energy saving under alternative process plans, J. Mech. Eng. 53 (5) (2017) 12–23.
[15] S. Zhao, Hybrid algorithm based on improved neighborhood structure for flexible job shop scheduling, Comput. Integr. Manuf. Syst. 24 (12) (2018) 3060–3072.
[16] C. Wang, M. Zhang, Z. Ji, Y. Wang, Genetic algorithm for solving multi-objective dynamic flexible job shop scheduling, J. Syst. Simul. 29 (08) (2017) 1647–1657.
[17] T. Mao, Research On Application of Hybrid Particle Swarm Algorithm in Flexible Job Shop Scheduling, Zhejiang University. 2018.
[18] M. Dai, D. Tang, A. Giret, M. Salido, Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints, Robot. Comput. Integr. Manuf. 59 (2019) 143–157.
[19] X. Liang, M. Huang, T. Ning, Flexible job shop scheduling based on improved hybrid immune algorithm, J. Ambient Intell. Humaniz. Comput. 9 (1) (2018) 165–171.
[20] X. Huang, X. Zhang, Y. Ai, ACO integrated approach for solving flexible job-shop scheduling with multiple process plans, Comput. Integr. Manuf. Syst. 24 (03) (2018) 558–569.
[21] M. Li, D. Lei, H. Xiong, An imperialist competitive algorithm with the diversified operators for many-objective scheduling in flexible job shop, IEEE Access 7 (2019) 29553–29562.
[22] L. Feng, T. Liao, Z. Lin, Two-level multi-task scheduling in a cloud manufacturing environment, Robot. Comput. Integr. Manuf. 56 (2019) 127–139.
[23] G. Zhang, L. Wu, L. Nie, Y. Wang, Robust flexible job shop scheduling method with machine breakdowns, J. Syst. Simul. 28 (04) (2016) 867–873.
[24] G. Gong, R. Chiong, Q. Deng, X. Gong, A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility, Int. J. Prod. Res. 58 (14) (2020).
[25] T. Jiang, Low-carbon workshop scheduling problem based on grey wolf optimization, Comput. Integr. Manuf. Syst. 24 (10) (2018) 2428–2435.
[26] Q. Zhang, Z. Yu, Discrete fruit fly optimization algorithm based on dominant population for solving no-wait flow shop scheduling problem, Comput. Integr. Manuf. Syst. 23 (03) (2017) 609–615.
[27] F. Luan, Z. Cai, S. Wu, Optimizing the low-carbon flexible job shop scheduling problem with discrete whale optimization algorithm, Mathematics 7 (8) (2019).
[28] Y. Lu, J. Lu, T. Jiang, Energy-conscious scheduling problem in a flexible job shop using a discrete water wave optimization algorithm, IEEE Access 7 (2019) 101561–101574.
[29] V. Roshanaei, A. Azab, H. Elmaraghy, Mathematical modelling and a meta-heuristic for flexible job shop scheduling, Int. J. Prod. Res. 51 (20) (2013) 6247–6274.
[30] M. Dai, D. Tang, A. Giret, Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm, Robot. Comput. Integr. Manuf. 29 (5) (2013) 418–429.
[31] H. Cao, J. Zhou, P. Jiang, K. Hon, Hao Yi, C. Dong, An integrated processing energy modeling and optimization of automated robotic polishing system, Robot. Comput.-Integr. Manuf. 65 (2020), 101973.
[32] Z. Wu, S. Hsieh, J. Li, Sensor deployment based on fuzzy graph considering heterogeneity and multiple-objectives to diagnose manufacturing system, Robot. Comput. Integr. Manuf. 29 (1) (2013) 192–208.
[33] W. Xia, Z. Wu, An effective hybrid optimization approach for multi-objective flexible job shop scheduling problems, Comput. Ind. Eng. 48 (2) (2015) 409–425.