



Low carbon emission management for flexible job shop scheduling: a study case in China

Tao Ning^{1,2} · Yiming Huang³

Received: 3 August 2020 / Accepted: 3 June 2021 / Published online: 14 June 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

For flexible job-shop scheduling problem with low carbon emission constraints, an improved quantum genetic algorithm based on double chains coding is proposed. Firstly, a mathematical model is established to minimize makespan, total workload of machines and carbon emissions of machines. Secondly, carbon emission equations in job shop scheduling process are inducted and designed. Based on the selected model, a method using an improved quantum genetic algorithm with double chains to solve processing route selection is proposed. Finally, on the basis of Kacem example, the performance of the method proposed in the paper was analyzed by ANOVA through experimental simulation and compared with the algorithms commonly used at present. The results show that the method not only achieves the goal of optimization, but also meets the practical requirements of reducing carbon emissions in production and processing.

Keywords Flexible job-shop scheduling · Carbon emission · Double chains quantum encoding · Improved quantum genetic algorithm

1 Introduction

Flexible job-shop scheduling has been proved to be a NP-hard problem, and the discharged CO₂ in the production has a serious impact on the environment. Studying low-carbon emission flexible job shop scheduling issues and realizing low-carbon green manufacturing is an important direction for flexible job shop scheduling and a necessary requirement for the benign development of manufacturing. According to the survey, the energy consumption of manufacturing industry accounts for nearly 1/3 of the global total consumption (Wang et al. 2018a). With the increasing awareness of energy conservation and emission reduction, the low-carbon scheduling of flexible job-shop has attracted the attention of scholars at home and abroad.

1.1 Related works

Because of the high complexity of the problem, many researchers use intelligent optimization algorithm to solve the problem, such as shuffled frog-leaping algorithm (Jiang 2018), drosophila algorithm (Wu and Sun 2018a, b), particle swarm optimization (Wu and Sun 2018a, b) and ant colony optimization (Piroozfard et al. 2018), etc., which have been applied to low-carbon scheduling. Quantum genetic algorithm (QGA) is a combination of quantum mechanics theory and genetic algorithm. Domestic scholars built high dimensional and low carbon scheduling model and optimized quantum genetic algorithm for low-carbon dispatch FJSP problem with multidimensional economic indicators and green indicators, which make up for the shortage of insufficient pressure in many-objective space selection (Ning et al. 2019a). Zheng and Ling (2018) extended the results of static job shop scheduling to dynamic flexible job shop scheduling, and proposed a dynamic scheduling algorithm based on hybrid quantum genetic algorithm. Ming and Deming (2019a, b) proposed an improved quantum genetic algorithm based on particle swarm algorithm to build a mathematical model of the problem by minimizing completion time, total cost and penalty value. Multi-objective and low carbon FJSP

✉ Tao Ning
daliannt@126.com

¹ Institute of Computer Science and Engineering, Dalian Minzu University, Dalian 116000, China

² Big Data Application Technology Key Laboratory of State Ethnic Affairs Commission, Dalian 116000, China

³ Institute of Software, Dalian Jiaotong University, Dalian 116021, China

is still in its infancy in China, but it has become a hot and difficult point in scheduling research. Wang et al. summarized the research progress on such problems in China from optimizing green shop scheduling, introducing quantum bits into genetic coding for the first time, and using quantum gates to update quantum chromosomes, which achieved good results (Wang et al. 2018b). From considering FJSP low carbon scheduling problem of depending on sequence release time, Li Ming et al. collated and improved the framework of quantum evolutionary algorithms, adding quantum variations and quantum crossover operations to optimize the total energy consumption of key and non key objectives. They analyzed the relationship between deterioration of total energy consumption and improvement of key objectives (Ming and Deming 2019a, b). Literature (Gong et al. 2019; Li et al. 2018) shows job shop scheduling model of minimize machine energy consumption. Compared with the scheduling that minimizes the completion time, this model effectively reduces the energy consumption of machine idling. In the literature (Shan et al. 2020; Caijie et al. 2020), carbon emission factors caused by energy consumption in shop scheduling problem is considered, and the overall carbon emission in the manufacturing process is reduced through an improved quantum cloud genetic algorithm. Literature (Ning et al. 2018a, b, 2019a, b, 2020) showed quantum genetic algorithm based on bacterial foraging algorithm to minimize completion times and total energy consumption for the production process for the optimization goal, which considered energy consumption during material cutting and normal operation of the machine, and meet the goal of saving energy for FJSP by optimizing environmental factors within peak power.

1.2 Contribution of this paper

Although relevant scholars have made effective research on the flexible job shop scheduling problem, the parameters of processing method, processing route, and cutting of workpieces in the study are static. How to achieve efficient energy saving, emission reduction and environmental pollution reduction is one of the urgent problems in manufacturing industry. Firstly, a multi-objective model is established to minimize makespan, total workload of machines and carbon emissions of machines. Secondly, an improved quantum genetic algorithm with double chains quantum encoding is used to solve the problems, and the validity and efficiency of carbon emission model are analyzed by ANOVA. At last, a new scheduling strategy is proposed in this paper, which can reduce adjustment time and waiting time of machines through changing the processing order of workpieces on each machine, moreover,

it can also reduce the energy consumption generated by machine idling.

2 FJSP model of low-carbon emission

2.1 Description of problem

The low-carbon emission FJSP problem is an extension of the FJSP problem. This kind of problem not only needs to solve the problem of processing machine selection and process sequencing in the production workshop, but also needs to achieve the goal of minimizing the carbon emissions of machine processing (Ning and Wang 2018). The low-carbon emission FJSP problem can be described as: The shop consists N processing workpieces and M machines. Each job i ($i \in \{1, 2, \dots, N\}$) has n_i ($n_i \geq 1$) processes, and each process can be produced on any machine with processing capability. R_{ij} represent j ($j \in \{1, 2, \dots, n_i\}$) process of workpiece i . $M_{ij} \subseteq \{1, 2, \dots, M\}$ are defined as the set of machines that can process the j operation of job i . Process R_{ij} can be processed in any machine m_k ($k = 1, 2, \dots, M_{ij}$) in M_{ij} . Machine m can process multiple processes for different workpieces (Ning et al. 2018a, b). The difference in machine performance makes the processing time and energy consumption of the process R_{ij} on the machine m different. The optimization goal of low-carbon scheduling is to allocate a suitable machine m for each process R_{ij} ; Meanwhile, the processing order of the workpieces on machine m are arranged and the start processing time is determined, so as to achieve a synergistic optimization of the efficiency indicators and the low-carbon emission indicators. In this paper, the following assumptions are made when solving the flexible job shop scheduling problem under low-carbon constraints:

- (1) A process in which the same machine can only process one workpiece at a certain time;
- (2) The same process can only be processed on one machine;
- (3) It is not allowed to be interrupted when the process is processed on the machine;
- (4) During the processing, if the machine does not break down, it cannot be shut down once it is started;
- (5) There is no priority order between processes of different workpieces, and there is a sequence relationship between processes of the same workpiece;
- (6) All workpieces can be processed at time $t = 0$, with the same priority;
- (7) Carbon emissions are also generated when the machine is idle after starting, and the carbon emissions generated when the machine processes the workpiece are independent of the type of workpiece.

2.2 Equation of carbon emission equation and related parameters

t_{ijk} , p_{ijk} , c_{ijk} represent the processing time, processing power and carbon emission of R_{ij} on machine m_k .

The carbon emission of job-shop mainly comes from the power consumption. Taking electricity consumption as a measure of carbon emission indicators, the coefficient matrix of carbon emission in this paper can be expressed as follows:

$$C = \{c_{ijk}\}_{m \times n} \tag{1}$$

The balance equation of power p can be expressed as follow:

$$p = p^e + p^c + p^w \tag{2}$$

In which, p^e is the no-load power to maintain its own running, p^c is the cutting power for workpiece processing and p^w is the load loss power to bear the processing load.

In the actual machining process, p^c is proportional to p^w , if the ratio is expressed as load loss power coefficient of λ , then Eq. (2) can be simplified as $p = p^e + (1 + \lambda)p^c$. Set the processing time as t_{ijk} ; the no-load power of m_k is p_{ijk}^e , the cutting power is p_{ijk}^c , and the carbon emission for R_{ij} on m_k can be expressed as:

$$c_{ijk} = \int_0^{t_{ijk}} p_{ijk}^e(t)dt + (1 + \lambda) \int_0^{t_{ijk}} p_{ijk}^c(t)dt \tag{3}$$

Some researches show that the cutting energy of the same workpiece on the same kind of machine is approximately constant (Kim et al. 2017) and the change of load loss power factor can be ignored. Therefore, the difference of carbon emission in the machine process mainly comes from the no-load power, and the carbon emission coefficient matrix of R_{ij} on m_k can be simplified as $c_{ijk} \approx \int_0^{t_{ijk}} p_{ijk}^e(t)dt$. While the no-load power of the machine mainly depends on the processing speed of the spindle, the carbon emission calculation can be further expressed as:

$$c_{ijk} \approx p_{ijk}^e(s) \times t_{ijk} \tag{4}$$

In Eq. (4) s is the parameter related to the processing speed.

2.3 Objective function

The goal of low-carbon FJSP scheduling is to select the appropriate machine for each process of the workpiece and determine the optimal processing order for each process of each machine, so as to minimize the carbon emissions of the processing machines in the workshop. In this paper, minimizing the average completion time of the workpiece, minimizing the total cost, and minimizing carbon emissions

are established. Considering the differences between multiple targets, the key to solve the problem lies in finding a satisfactory balanced solution among the multiple targets. The objective function is established as follows:

Minimize makespan

$$f1 = \min(F) = \min \left[\left(\sum_{k=1}^M F_{m_k} \right) / M \right] \tag{5}$$

$$F_{m_k} = \sum_{i=1}^N \sum_{j=1}^{n_i} (S_{ijk}b_{ijk} + S_{ijk}t_{ijk}) \tag{6}$$

Minimize carbon emissions

$$f2 = \min(C) = \min \left(\sum_{i=1}^N \sum_{j=1}^{n_i} \sum_{k=1}^M t_{ijk}P_{ijk}S_{ijk} \right) \tag{7}$$

Minimize total cost

$$f3 = \min(A) = \min \left[\sum_{i=1}^N \left(A_i + \sum_{j=1}^{n_i} \sum_{k=1}^M A_{ijk}S_{ijk} \right) \right] \tag{8}$$

$$A_{ijk} = (\mu_{ijk} + v_{ijk}) \tag{9}$$

In the Eq. (5), F represents the completion time of all machines, as an important indicator to measure the workload of the machine; F_{m_k} represents the total completion time of machine m_k , b_{ijk} represents the starting processing time of process R_{ij} on machine m_k , t_{ijk} represents the processing time of process R_{ij} on machine m_k , and S_{ijk} represents the processing of process R_{ij} on machine m_k in Eq. (6), which means:

$$S_{ijk} = \begin{cases} 1, & R_{ij} \text{ is cut on } m_k; \\ 0, & R_{ij} \text{ is not cut on } m_k. \end{cases}$$

In Eq. (7), C and p_{ijk} represent the total carbon emissions and processing power of the process R_{ij} processed on the machine m_k respectively. In Eq. (8), A represents the total processing cost of workpiece i , A_i represents the raw material cost of workpiece i , and A_{ijk} represents the cost of processing R_{ij} on machine m_k . In Eq. (9), μ_{ijk} and v_{ijk} represent the labor cost and machine cost of processing R_{ij} on machine m_k respectively.

2.4 Constraints conditions

(1) Operation constraints

There are constraints between processes of the same workpiece, that is, the j th process of the workpiece i must start after the completion of $(j - 1)$ th process.

$$\sum_{k=1}^M b_{ijk} S_{ijk} \geq \sum_{k=1}^M [(b_{i(j-1)k} t_{i(j-1)k})] S_{i(j-1)k} \quad (10)$$

In Eq. (10), $S_{ijk} = S_{i(j-1)k} = 1$.

(2) Machine constraints

The same machine can only processed one process at the same time, that is, for process R_{ij} at time $t(t > 0)$, if $\exists S_{ijk} = 1$, then $S_{xyk} = 1$ must not be established (when $i = x, j \neq y$).

(3) Continuity constraints

The process R_{ij} cannot be interrupted during processing:

$$f_{ijk} = \begin{cases} \max\{f_{i(j-1)k}, b_{ijk}\} + t_{ijk}, & j > 1; \\ b_{ijk} + t_{ijk}, & j = 1. \end{cases} \quad (11)$$

In Eq. (11), f_{ijk} indicates the completion time of the process R_{ij} .

3 Improved quantum genetic algorithm

An improved quantum genetic algorithm is established in this paper to solve the multi-objective FJSP low-carbon scheduling problem. First, the double-chain genetic algorithm is used to encode the machine distribution chain and the process ranking chain. Secondly, the non-dominated Pareto solution ranking method is used to rank the population. The evolutionary population is selected by calculating the crowding distance (Sheremetov et al. 2018).

3.1 Encoding and decoding

In this paper, a double-chain coding method based on the combination of machine chain and process chain is used to determine machine selection and process ordering. Based on the equal coding length of the machine and the total number of processes of all the workpieces, the chromosomes are sorted from left to right according to the number of the workpieces and processes. The coding length of the processes is equal to the total number of processes of all the workpieces, and the processes of each workpiece are labeled with the corresponding workpiece number. The integer at the locus represents the sequential number of the current processing machine in the set of optional machines, so the integer on the locus represents the sequential number of the current processing machine in the set of optional machines.

The insertion greedy decoding algorithm is used for chromosome decoding to ensure the generation of active scheduling solutions. The process of chromosomes reads in sequence, and the processes are inserted into the available machine idle interval to reduce machine idle operation,

thereby improving machine utilization and shortening the maximum completion time.

3.2 Quantum genetic algorithm

Quantum genetic algorithm (QGA) is a combination of quantum evolution algorithm and genetic algorithm. The chromosome coding of genetic algorithm is replaced to quantum bit probability amplitude by quantum bit, quantum coding and quantum overlay state. A quantum bit state can be expressed as $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$. Where α and β are plurals, corresponding to the probability amplitudes of $|0\rangle$ state and $|1\rangle$ state respectively, and $|\alpha|^2 + |\beta|^2 = 1$, in which the chromosome with n quantum bit coding can be represented 2^n states. The key of quantum genetic algorithm is to add quantum crossover and quantum variation after quantum rotatory gate operation. Although this can avoid trapping into local minima to a certain extent, the possibility of trapping into local minima is not completely eliminated. Literature (Zhu et al. 2020) showed a method for dynamically adjusting the quantum rotating gate as follows:

$$\begin{bmatrix} \alpha_i^T \\ \beta_i^T \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (12)$$

$[\alpha_i, \beta_i]^T$ represents the i th qubit of chromosome, θ_i represents the rotation angle, and its size and symbol are determined by the following equation:

$$\theta_i = D(\alpha_i, \beta_i) \Delta \theta_i \quad (13)$$

The symbol $\Delta \theta_i$ in Eq. (13) represents the angular step size of the rotation angle, and the symbol $D(\alpha_i, \beta_i)$ represents the rotation direction of the rotation angle. QGA searches use quantum bit probability amplitude to represent chromosome coding and quantum gate updating, which guarantees the diversity and parallelism of population. In this paper, qubits are used to store and express quantum chromosome. Quantum chromosome can be “0” state or “1” state, or any superposition state of them. That is, what the quantum chromosome expresses is no longer certain information. It contains all possible information, and any operation on quantum chromosome will also affect all possible information at the same time. In this paper, the use of quantum chromosome allows the genetic individual to select quantum chromosome in multiple superposition states, so that the improved quantum genetic algorithm has better diversity characteristics. However, QGA is complex in coding and decoding when it is used to solve multi-objective complex optimization problems, so an improved double chains quantum genetic algorithm (IDCQGA) is proposed in this paper.

3.3 Improved double chains quantum genetic algorithm

3.3.1 Double chains quantum coding

In the genetic operation, the estimated distance between the current solution and the target optimal solution is corrected by the compensation factor. The ability of optimizing quantum genetic algorithm is enhanced, the occurrence of inferior value is reduced and the errors between the current solution and the optimal solution are corrected. In the genetic operation, the positioning distance between the current solution and the optimal solution is calculated, and the compensation factor is determined. The algorithm is continuously corrected by the compensation factor, and the population with large positioning error is eliminated. This helps to improve the positioning accuracy of the solution in the algorithm. Moreover, it can make algorithm find better solutions, and effectively improve the convergence performance. The correction process of the compensation factor is shown in the following Fig. 1.

Therefore a new compensation factor γ ($\gamma \geq 1$) on the basis of probability amplitude coding is introduced in this paper, assuming that cr_i represents a quantum chromosome. Then the i th chromosome coding scheme is as follows:

$$cr_i = \begin{bmatrix} \alpha_{i1} & \alpha_{i2} & \dots & \alpha_{im} \\ \beta_{i1} & \beta_{i2} & \dots & \beta_{im} \end{bmatrix} = \begin{bmatrix} \cos(\gamma t_{i1}) & \cos(\gamma t_{i2}) & \dots & \cos(\gamma t_{im}) \\ \sin(\gamma t_{i1}) & \sin(\gamma t_{i2}) & \dots & \sin(\gamma t_{im}) \end{bmatrix} \tag{14}$$

Among them, α, β need to meet the normalization constraint, that is $|\alpha|^2 + |\beta|^2 = 1$; $t_{ij} = 2\pi \times \vartheta$, ϑ represents a random number between (0, 1); $i = 1, 2, \dots, u$; $j = 1, 2, \dots, v$; u represents the population size, v represents the number of

qubits. The compensation factor γ extends the period from 2π to multi-period, which can improve the convergence probability of the algorithm. Each chromosome contains two parallel gene chains. These two gene chains respectively represent the machine distribution chain and process sequence chain of the FJSP problem, which means that each chromosome corresponds to two optimal solution, that is:

$$cr_{i\cos} = (\cos(t_{i1}), \cos(t_{i2}), \dots, \cos(t_{im}))$$

$$cr_{i\sin} = (\sin(t_{i1}), \sin(t_{i2}), \dots, \sin(t_{im}))$$

$cr_{i\cos}$ and $cr_{i\sin}$ are called cos and sin solutions, respectively. When the chromosome is iterated, the two solutions can be updated synchronously.

3.3.2 Non-dominated optimal sorting

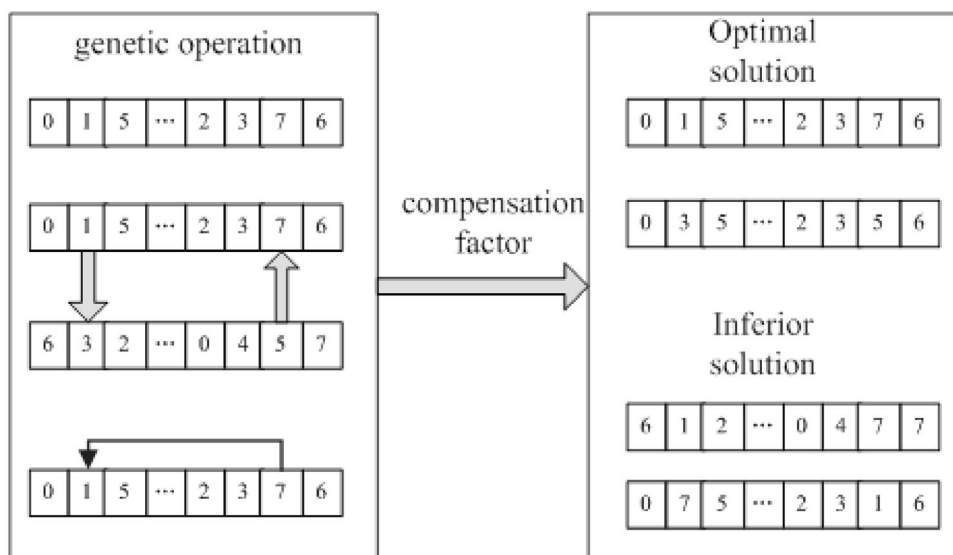
The solution of FJSP problem is difficult to obtain the optimal solution satisfying all the goals. In this paper, a non-dominated Pareto solution sorting method is proposed based on the fuzzy set theory. In this method, the parameters Z_g and z_g of individual g in population u are calculated to achieve classification. The specific steps are as follows:

Step 1: Initialize the parameter set Z_g , which includes all the individuals dominated by the individual g , so that $Z_g = \emptyset$;

Step 2: Initialize the parameter variable z_g , which represents the total number of individuals that can dominate the individual g ;

Step 3: To calculate the dominance relationship, set $g, g' \in u$, if g can dominate g' , then make $Z_g = Z_g \cup \{g'\}$; if g' dominate g , then $z_g = z_g + 1$; if $z_g = 0$, then g is a non-dominated individual, which belongs to the first level denoted as $g' = 1$, makes g join the rank set R_1 , that is $R_1 = R_1 \cup \{g\}$;

Fig. 1 The influence of the compensation factor



Step 4: Let G' represent the set of remaining individuals g' , let $i = 1$, when $R_i \neq \emptyset$, $G' = \emptyset$, for individuals $g' \in Z_g$, let $z_g = z_g - 1$, if $z_g = 0$, then $g'' = i + 1$; let $G' = G' \cup \{g'\}$, $i = i + 1$, $R_i = G'$;

Step 5: Judge whether R_i is empty. If it is empty (all individuals have corresponding levels), stop; otherwise, go to Step 4.

Corresponding pseudocode:

Begin

Input individual g

Initialize the parameter set Z_g , and $Z_g = \emptyset$.

Initialize the parameter variable z_g ,

Set $g, g' \in u$,

if g can dominate g' then

make $Z_g = Z_g \cup \{g'\}$.

else if g' dominate g then

do $z_g = z_g + 1$;

if $z_g = 0$ then

output g is a non-dominated individual and g belongs to

The first level,

denoted as $g' = 1$

makes g join the rank set R_1 and $R_1 = R_1 \cup \{g\}$.

for ($i = 1$, R_i is empty, $i = i + 1$)

Let G' represent the set of remaining individuals g' .

if $G' = \emptyset$ and $g' \in Z_g$ then

$z_g = z_g - 1$,

if $z_g = 0$ then

do $g'' = i + 1$.

let $G' = G' \cup \{g'\}$.

$R_i = G'$.

End

output all individuals have corresponding levels

End

The smaller the non-dominated sorting value, the better the individuals are, if the non-dominated sorting value is the same, the individual with larger crowding distance will be preferred.

3.3.3 Selection strategy based on crowded distance

The crowding distance of a chromosome individual is obtained by calculating the sum of the distance difference between two adjacent chromosomes in the same rank on each sub target (Reddy et al. 2018). First, all the chromosomes at the same rank are sorted in ascending order according to the fitness value of target j . Assuming that the crowding distance of the first and last chromosomes is infinite, the crowding distance of chromosome target j between them is:

$$L[g]_{dj} = \frac{f_j^{g+1} - f_j^{g-1}}{f_j^n - f_j^1} \quad (15)$$

In Eq. (15), $L[g]_{dj}$ represents the crowding distance of target j of chromosome g , f_j^x represents the fitness value of the j th target of the x th chromosome in the current rank, where the values of x are 1, $g - 1$, $g + 1$, n . The crowding distance of the chromosome is the sum of all target crowding distances of the current chromosome $L[g]_{dj}$. The chromosomes with large crowding distance are more likely to be selected for evolution, which is conducive to maintaining population diversity (Shao et al. 2013). By calculating the crowded distance, the algorithm can converge to evenly distributed Pareto surfaces.

The non-dominated sorting and crowding distances make each chromosome individual g obtain two attributes: the non-dominated sorting rank g' and the crowding distance $L[g]_{dj}$. Thus, the partial order relationship $>_n$ is defined as $g >_n g'$: if $g' < g''$ or $g' = g''$ and $L[g]_d > L[g']_d$; The meaning of this partial order relationship is that the solution g is better than the solution g' , and the partial order relationship is the basis of the selection operation.

3.3.4 Description of algorithm

The steps to optimize the double-chain QGA algorithm:

Step 1: The initial population u_0 is made up of n chromosomes by double-stranded quantum coding;

Step 2: Decoding chromosomes follow the restrictions;

Step 3: Population is processed by non-dominant sorting and calculation of crowding distance;

Step 4: Two individuals were randomly selected for a non-inferiority level comparison. If the levels are different, select the individuals with lower levels; otherwise, select the individuals with low crowding density to generate a new population u_1 and ensure that all states appear with the same probability in the early algorithm search;

Step 5: Solve suitability values of each chromosome, record the contemporary optimal solution X_a and the corresponding quantum chromosome g_a , and the current optimal solution X_b and the corresponding quantum chromosome g_b ;

Step 6: The quantum bit corresponding to the current optimal chromosome g_b are the target to determine the size and direction of the quantum gate rotation angle, and update the quantum bit according to the quantum gate;

Step 7: Each chromosome is mutated by the quantum Hadamard gate according to the probability of mutation;

Step 8: Determine whether the convergence condition is satisfied or the maximum number of iterations has been reached. If it is satisfied, go to Step 9; otherwise, the iteration number add 1 and go to Step 2;

Step 9: Output current optimal solution X_b and corresponding fitness suitability values.

Corresponding pseudocode:

Begin

n chromosomes by double-stranded quantum coding
 Input the initial population u_0
 for (iteration = 0, the convergence condition is satisfied or the maximum number of iterations has been reached, the iteration number add 1)
 Decoding the chromosomes follow the restrictions
 Population is processed by non-dominant sorting
 calculation of crowding distance;
 Select two individuals randomly to compare the non-inferiority ranks.
 If the levels are different then
 select the individuals with lower levels;
 else if then
 select the individuals with low crowding density to generate a new population u_1
 record the contemporary optimal solution X_a
 record the corresponding quantum chromosome g_a
 record the current optimal solution X_b
 record the corresponding quantum chromosome g_b ;
 if quantum bit corresponding to the current optimal chromosome g_b then
 determine the size and direction of the quantum gate rotation angle
 update the quantum bit according to the quantum gate
 each chromosome is mutated by the quantum Hadamard gate
 End
 Output current optimal solution X_b and corresponding fitness suitability values.
 End

3.3.5 Sensitivity analysis

- (1) The influence of multi-workpiece issues
 In order to show the influence of problem complexity on the factual response ability of the decision-making method, Fig. 2 shows the comparison of real-time processing time and decision space of a single sample in Brandimarte examples Mk01/Mk02/Mk03/Mk04 (Brandimarte 2011) for single-workpiece machining and multi-workpiece machining problems. It can be seen from Fig. 2 that the processing time and decision space increase exponentially when the single-workpiece problem is extended to multi-workpiece problem.
- (2) The influence of changes in processing demand
 In order to show the influence of changes in processing demand on the algorithm in this paper and carbon emission cost, Fig. 3 takes the example MK03 (15 × 8) as an example to show the variation curve of the carbon emission cost of 1000 online samples with the dynamic demand under different algorithms. It can be seen from Fig. 3 that no matter what algorithm is used, the system needs to pay high cost to deal with high dynamic

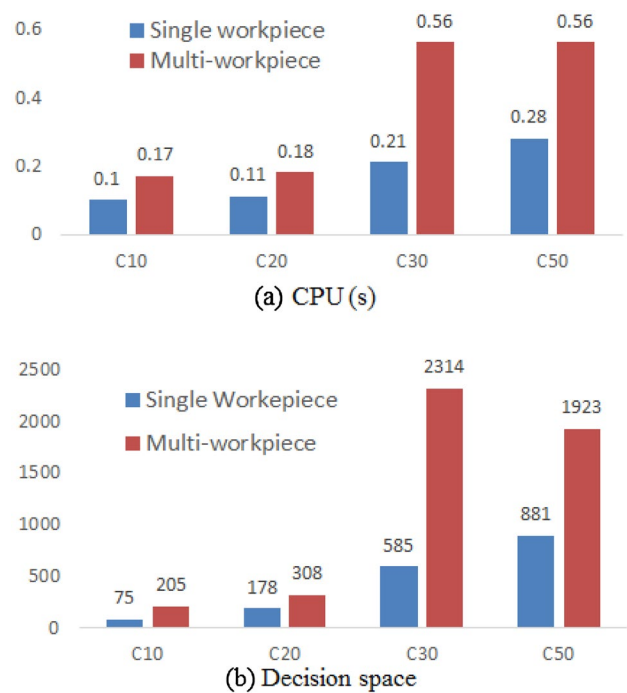


Fig. 2 Comparison on the CPU and decision space for a single sample

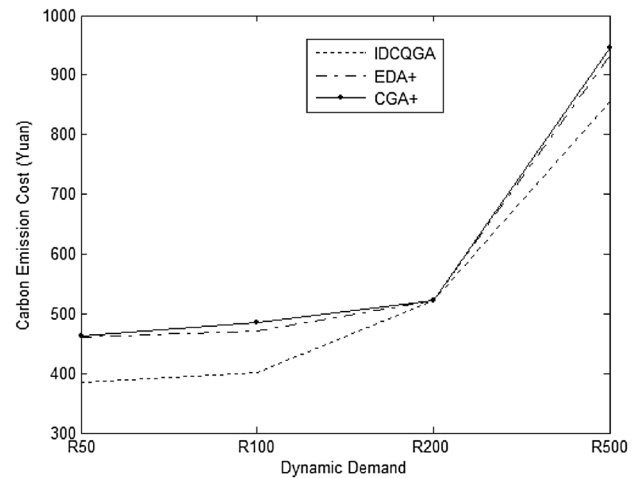


Fig. 3 Carbon emission cost curve and of three algorithms under different dynamic demand

demand. However, according to the variation curve, the cost increase of IDCQGA proposed in this paper is less than that of EDA+ (estimation of distribution algorithm) (Zeng et al. 2018) and CGA+ (canonical genetic algorithm) (Yang et al. 2019), which indicates that with the increase of demand uncertainty, the system will inevitably produce high carbon emission cost. But in general, IDCQGA is superior to EDA+ and CGA+ when demand degree changes dynamically.

(3) The influence of α and β

Analyze the effect of improved algorithm data on model results, and take different α and β for simulation analysis (Ning et al. 2018a, b). The optimal comprehensive cost and carbon tax cost results under different algorithm parameters are shown in Table 1.

In Table 1, five different values of the proposed algorithm parameters are taken respectively, which have little effect on the overall cost and carbon emission cost of FJSP, and the experiment results are relatively stable.

Table 1 Comparison of optimal results under different algorithm parameters

(α, β)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)
Optimal cost Z	4435	4404	4393	4418	4407
Carbon tax cost	4407	4105	3978	4272	4113

Table 2 Data of workpiece

No.	Parent work-piece	Name	Code	No. of processing	Mission	Work hours (m)	Department
1	–	Cardan shaft	TF022000-87	1	Assemble	50	Assembly one
2	1	Balance block assembly	TF022000-87	1	Assemble	20	Assembly one
3	1	Flange fork assembly	TF022013/012-87	1	Assemble	20	Assembly one
4	1	Sliding fork assembly	TF022008/012-87	1	Assemble	20	Assembly one
5	1	Spline shaft fork assembly	TF022011/012-87	1	Assemble	20	Assembly one
6	1	End cover	TF022007-87	1	Assemble	25	Assembly two
7	1	Anti-off nut	TF022009-87	1	Assemble	15	Assembly one
8	1	Lining tile	TF022010-87	1	Roughcast	0	Processing second class
				2	Rough car	8	Processing second class
				3	Milling	18	Processing second class
				4	Finishing car	20	Processing class
				5	Cleanup. Lapping. Assembly welding	10	Small link class
9	4	Sliding fork	TF022002-87	1	Roughcast	0	Processing second class
				2	Rough car	40	Processing second class
				3	Mark(1)	8	Processing class
				4	Boring and milling inside	30	Processing second class
				5	Drill	2	Processing class
				6	Boring	12	Processing second class
				7	Quenching and tempering	35	Heat treatment class
				8	Finishing car	50	Processing class
				9	Mark(2)	6	Processing class
				10	Tensile spline	55	Processing second class
				11	Grinding process surface	26	Processing second class

4 Validation and analysis of experiment

The application research object of this case is the bogie of Dongfeng 4D diesel locomotive, including shock absorber seat, casing, derrick, traction rod and other parts, with codes TF022000-TF022010, and case data is shown in Table 2. The workshop has 13 operation centers, 29 assembly carriers and 82 processes, including bogie fitter shift, heat treatment, small connecting rod shift and machining shift, etc. The delivery date of each batch of processing parts is different, which belongs to the discrete production and processing mode. Research and analysis found that the workshop has the following problems:

- (1) Although workshop production is carried out according to the priority of the workpiece, workshop production may be interrupted in the case of emergency insertion of the workpiece or the arrival of the workpiece, thus reducing the stability of scheduling.
- (2) The processing machine load in the workshop is uneven. FJSP allows parts to be processed in one of the machines with processing capabilities, but if the pro-

duction scheduling plan remains unchanged, then some machines will slowly appear to be overloaded. While the idle rate of other parts of the machine is high.

- (3) The parts such as shock absorber seat and casing processed in this workshop can be processed on multiple machine tools, and the processing time varies with different processing machines and operators, which is a typical FJSP.

4.1 Comparison of the algorithms result

In order to verify the performance of the proposed method, this paper aims at minimizing the average completion time f_1 , minimizing carbon emissions f_2 and minimizing the total cost f_3 . Taking the classic Kacem (Caijie et al. 2020) as an example, the design population size is 200, if the population size is too large, the running time will be lengthened, and the convergence effect will be unstable; if the population size is too small, it will obviously appear inbreeding and produce sick gene, which makes the population evolution unable to produce the expected number according to the pattern theorem. Experiments in literature (Ning et al. 2020) show that if the population number is 200, the convergence speed will be fastest and the performance of the algorithm will be the best. If the number of iterations is too small, the algorithm is not easy to converge, and the population is not mature; if the number of iterations is too large, the algorithm is already skilled or the population is too early to converge again, so it is meaningless to continue evolution, which will only increase the time expenditure and resource waste. When the number of iterations is 100 (Caijie et al.

2020), the running time is the shortest and the convergence speed is the fastest, the maximum iteration number is 100. If the crossover probability is too small, the population can not be effectively updated. If the crossover probability is too large, the randomness increases. With the increase of the crossover probability, the convergence is not necessarily fast or slow, but the time cost will increase. When the crossover probability is 0.45, the convergence situation is better (Ning et al. 2020). If the mutation probability is too small, the diversity of the population will decrease too fast, which will easily lead to the rapid loss of effective genes and difficult to repair; if the mutation probability is too large, although the diversity of the population can be guaranteed, the probability of high-order mode being destroyed will also increase. When the mutation probability is 0.02, the convergence effect and optimal value are the best, and the time is better than the mutation probability of 0.01 (Ning et al. 2020).

Using IDCQGA to solve five kinds of scale standard problems of Kacem ($4 \times 5, 8 \times 8, 10 \times 7, 10 \times 10, 15 \times 10$), and compared with the existing problems including CGA⁺, PSO⁺ (particle swarm optimization) (Mishra and Shrivastava 2018) and EDA⁺ and the data of comparison result is shown in Table 3. In Table 3, n represents the number of jobs, m represents the number of machines; rs_x ($x=1, 2, 3, 4$) represents different solutions obtained by the algorithm; T_l represents the maximum completion time of the machine (unit: min); A_t represents the cost of the job (unit: CNY); C_t represents the total carbon emission of the machine (unit: kg). It can be seen from the results of solving five Kacem examples with different algorithms in Table 3. The IDCQGA proposed in this paper can obtain more non-dominated solutions, and the current optimal solutions are obtained in the examples. Taking the 10×7 problem as an example,

Table 3 Comparison of Kacem of different algorithms

$n \times m$	Obj	CGA ⁺		PSO ⁺		EDA ⁺			IDCQGA			
		rs_1	rs_2	rs_1	rs_2	rs_1	rs_2	rs_3	rs_1	rs_2	rs_3	rs_4
4×5	T_l	15		10		10	11	12	10	10	11	10
	A_t	33		31		30	30	32	30	29	30	31
	C_t	45.2		45.2		45.2	40.6	38.4	45.2	40.6	40.6	40.6
8×8	T_l	14	15	14	14	13	14	15	13	14	14	13
	A_t	78	74	76	74	75	74	72	74	74	72	74
	C_t	62.5	62.5	60.7	60.8	60.8	60.7	62.5	60.4	60.2	62.5	59.8
10×7	T_l					11	10	11	11	10	10	
	A_t					60	60	59	59	61	59	
	C_t					55.4	58.2	58.2	55.1	50.6	58.2	
10×10	T_l	6		6		7	5	6	6	5	6	5
	A_t	44		42		40	41	40	40	39	40	40
	C_t	25.6		28.2		28.2	25.4	25.4	28.2	25.5	25.4	28.2
15×10	T_l	22		10		10	10		9	9		
	A_t	94		92		89	90		89	90		
	C_t	52.3		52.1		51.8	51.8		51.3	48.8		

although the EDA⁺ and IDCQGA algorithms both get three non-dominated solutions: (11, 60, 55.4), (10, 60, 58.2), (11, 59, 58.2) and (11, 59, 55.1), (10, 61, 50.6), (10, 59, 58.2), but the solution of EDA⁺ algorithm (11, 60, 55.4) is dominated by the solution of IDCQGA algorithm, the solutions (11, 59, 58.2) of EDA⁺ algorithm are dominated by the solutions (11, 59, 55.1) and (10, 59, 58.2) of IDCQGA algorithm.

4.2 Performance test of algorithm

In order to test the performance of the improved IDCQGA in solving the low-carbon FJSP, MATLAB R2010b was used in the experiment, and the platform has 2.8 GHz Intel Core I6 CPU and 4 GB RAM. The operation termination condition is up to 200 s. In order to reflect the influence of parameter optimization on carbon emission and maximum completion time, the parameters in literature (Sheremetov et al. 2018) are used in the experiment (as is shown in Table 4).

For the above example, CGA⁺, EDA⁺ and IDCQGA were tested and run 50 times respectively. For the three algorithms, after obtaining the optimal scheduling plan for each test, the maximum completion time, maximum machine load, carbon emissions, and weighted targets corresponding to the plan were calculated respectively. The average value and standard deviation of the obtained results are shown in Table 5, and the figures of mean value and SD are shown in Figs. 4 and 5.

It can be seen from Table 5 that in 50 experiments, the standard deviation of the maximum load of IDCQGA is slightly higher than the standard deviation of EDA⁺, but in comparison, the average and standard deviation of IDCQGA completion time are 7.9% and 55.3% better than CGA⁺. Compared with EDA⁺, it is optimized by 1.79% and 3.2%. The average and standard deviation of IDCQGA carbon emissions are 3.1% and 79.4% better than CGA⁺. Compared with EDA⁺, it is optimized by 0.8% and 56.7%. The mean and standard deviation of IDCQGA maximum load are 5.1% and 48.6% better than CGA⁺. The mean and standard deviation of the IDCQGA weighted target are optimized by 8.9% and 54.5% compared to CGA⁺, and the mean is optimized by 5.1% compared to EDA⁺, but the standard deviations of the three are the same. Therefore, for the three algorithms,

Table 5 The mean value and standard deviation of each scheduling target

Algorithm	Objective	Average value	SD
CGA ⁺	Makespan (m)	319.172	12.735
	Carbon emission (kg)	228.502	2.788
	Maximum load (m)	245.724	6.202
	Weighted objective (%)	67.90	0.011
EDA ⁺	Makespan (m)	299.398	5.886
	Carbon emission (kg)	223.217	1.325
	Maximum load (m)	233.392	3.002
	Weighted objective (%)	65.20	0.005
IDCQGA	Makespan (m)	294.053	5.698
	Carbon emission (kg)	221.364	0.574
	Maximum load (m)	233.163	3.185
	Weighted objective (%)	61.89	0.005

IDCQGA has the best solution. The algorithm not only has better targets, but also has the smallest standard deviation. It shows that the efficient and improved IDCQGA algorithm designed in the study has the best convergence and stability in solving FJSP for low-carbon optimization.

The results of the analysis of variance for each target obtained by the three algorithms are shown in Tables 6, 7, 8 and 9 and the corresponding figures can be seen in Figs. 6, 7, 8 and 9:

As can be seen from Tables 6, 7, 8 and 9, there are four goals to be considered: maximum completion time, carbon emissions, maximum machine load, and weighted goals. The solution performance of different algorithms is statistically significant, that is, Sig < 0.05. The comparison of the three algorithms shows that the solution performance and robustness of the IDCQGA algorithm are better than CGA⁺ and EDA⁺.

4.3 Comparison with other quantum genetic algorithm

In order to test the performance of IDCQGA and its improved strategy for solving low carbon flexible job shop scheduling, the performance of IDCQGA was compared with the popular classical quantum genetic algorithm of

Table 4 Parameter setting of different algorithms

IDCQGA		CGA ⁺		EDA ⁺	
Parameter	Value	Parameter	Value	Parameter	Value
POP size	200	POP size	200	POP size	200
PR of selection	0.65	Gravitational coefficient	95	Dominant POP rate	11%
PR of cross	0.20	Adjustment coefficient	2 × uptime/total elapsed time	Learning rate	0.2
PR of mutation	0.04	\	\	\	\

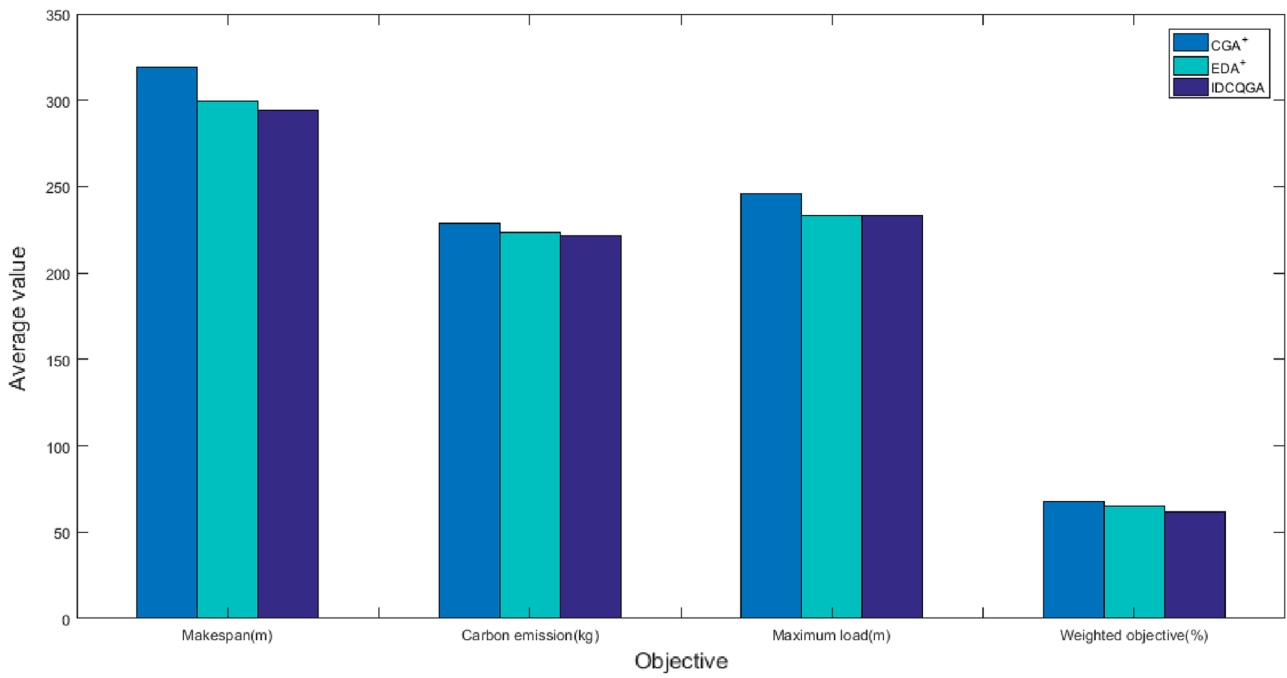


Fig. 4 The comparison of mean value of three algorithms

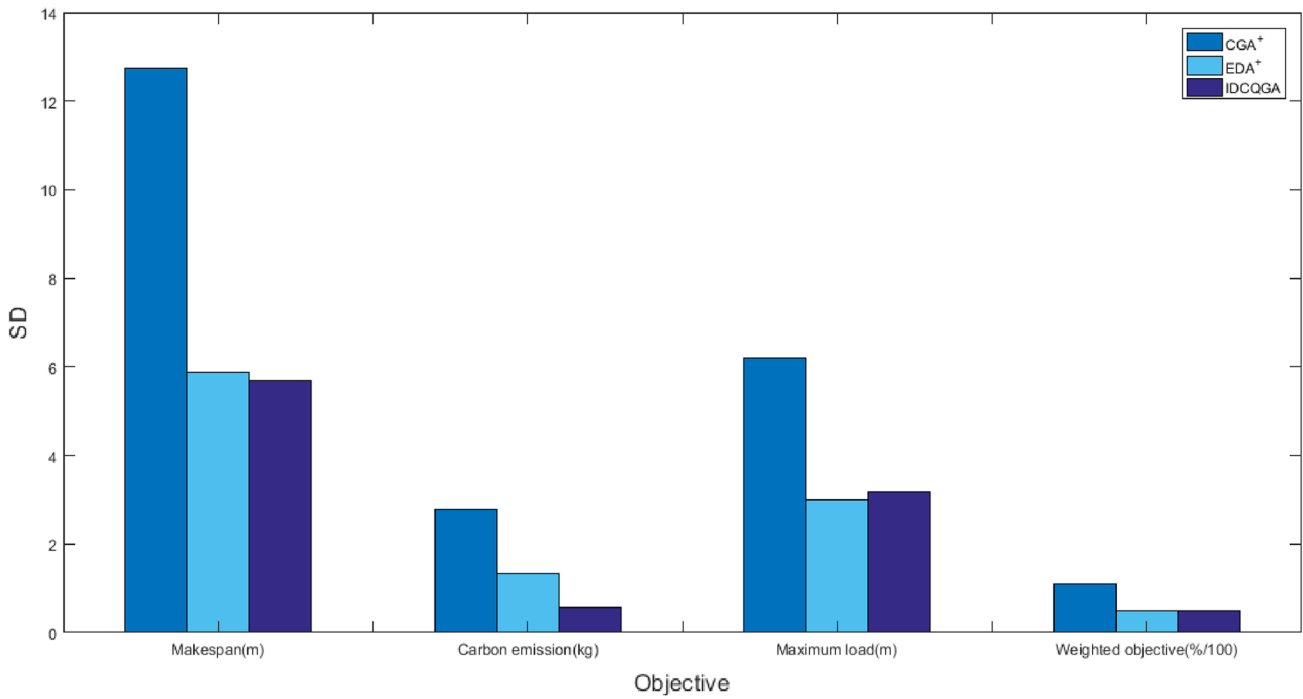


Fig. 5 The comparison of standard deviation of three algorithms

HQGA (Jin et al. 2020). In this section, MATLAB R2010a was used to run the computer with 2.8 GHz Intel Core I5 CPU and 4 GB RAM. The algorithm was set to terminate when the running time reached 900 s. The algorithm

parameters were obtained from the literature (Ning et al. 2020) (Psize = 80), the dominant population ratio was 10%, and the learning rate was 30%. The specific workpiece and lathe parameter data are shown in Table 2.

Table 6 ANOVA analysis of maximum makespan under different algorithms

	Sum of squares	Expectation function	Mean square	F-test	Sig
Between groups	560.780	2	279.800	8.059	0.001
Within groups	3018.300	87	34.650		
Total	3579.080	89			

Table 7 ANOVA analysis result of carbon emission under different algorithms

	Sum of squares	Expectation function	Mean square	F-test	Sig
Between groups	816.509	2	407.638	117.338	0.000
Within groups	301.682	87	3.479		
Total	1118.191	89			

Table 8 ANOVA analysis result of maximum load of machine under different algorithms

	Sum of squares	Expectation function	Mean square	F-test	Sig
Between groups	4278.253	2	2139.204	107.273	0.000
Within groups	1733.938	87	19.838		
Total	6012.191	89			

Table 9 ANOVA analysis result of weighted targets under different algorithms

	Sum of squares	Expectation function	Mean square	F-test	Sig
Between groups	0.023	2	0.013	196.891	0.000
Within groups	0.006	87	0.000		
Total	0.029	89			

In Matlab, the frontier solutions obtained by the algorithm are shown in Table 10 and in the three-dimensional diagram composed of three objects (eg Fig. 10). There are ten real frontier solutions obtained by the synthesis of IDCQGA and HQGA, that is, on the basis of the nine solutions of IDCQGA, plus which obtained by HQGA of [291 228.08 225], the details are shown in Table 11.

It can be seen from Tables 10, 11 and Fig. 10 that all the nine frontier solutions obtained by IDCQGA enter the real frontier solutions, while only two of the seven frontier solutions obtained by HQGA enter the real frontier solutions. Therefore, the performance of the multi-objective IDCQGA algorithm can be improved through the double-stranded quantum algorithm and the introduction of compensation factor, which not only obtains more frontier solutions, but also improves the quality of frontier solutions.

On the basis of the above testing, the classical Solomon (Piroozfard et al. 2018) examples are used to compare IDCQGA with other variations of quantum genetic algorithm. There are six kinds of examples in Solomon, and one standard problem from each kind of examples is selected and executed for 20 times using IDCQGA. The efficiency will be verified through the comparison with existing QGA-SVR (quantum genetic algorithm-support vector regression) (Deguang et al. 2020), ICQGA (improved cloud quantum genetic algorithm) (Yansong et al. 2020), CGA⁺ and HQGA. To verify the performance of the IDCQGA proposed in

this paper. In the experimental environment set in this paper, benchmark tests (CEC2017 test function) are used to compare the above three algorithms independently executed 20 times to verify the feasibility and effectiveness of the proposed algorithm. The benchmark test results are shown in Fig. 11.

This paper uses the benchmark test function (CEC2017) to test the performance of the algorithm proposed in this paper, which can have better algorithm performance evaluation indicators. At the same time, the test method is suitable for machine learning, combinatorial optimization, FJSS and other problems. At the same time, the FJSS problem designed in this paper is a non-linear programming problem, which can be better adapted to the test function. It can be seen from Fig. 11 that the IDCQGA proposed in this paper converges to the optimal

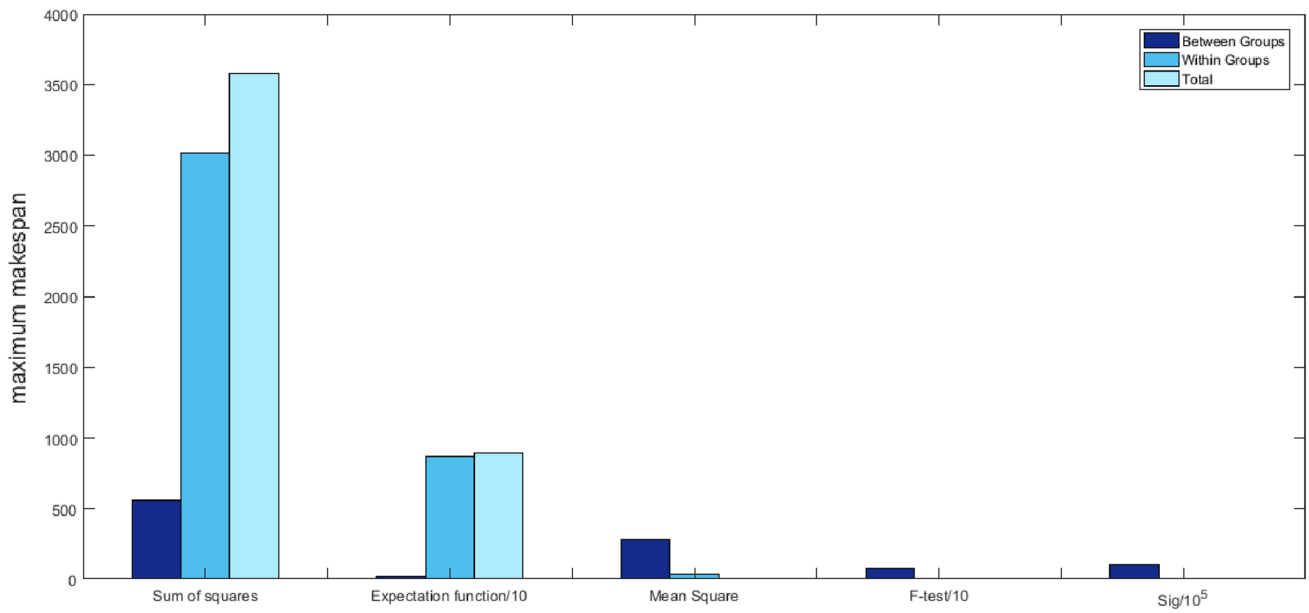


Fig. 6 ANOVA analysis of maximum makespan

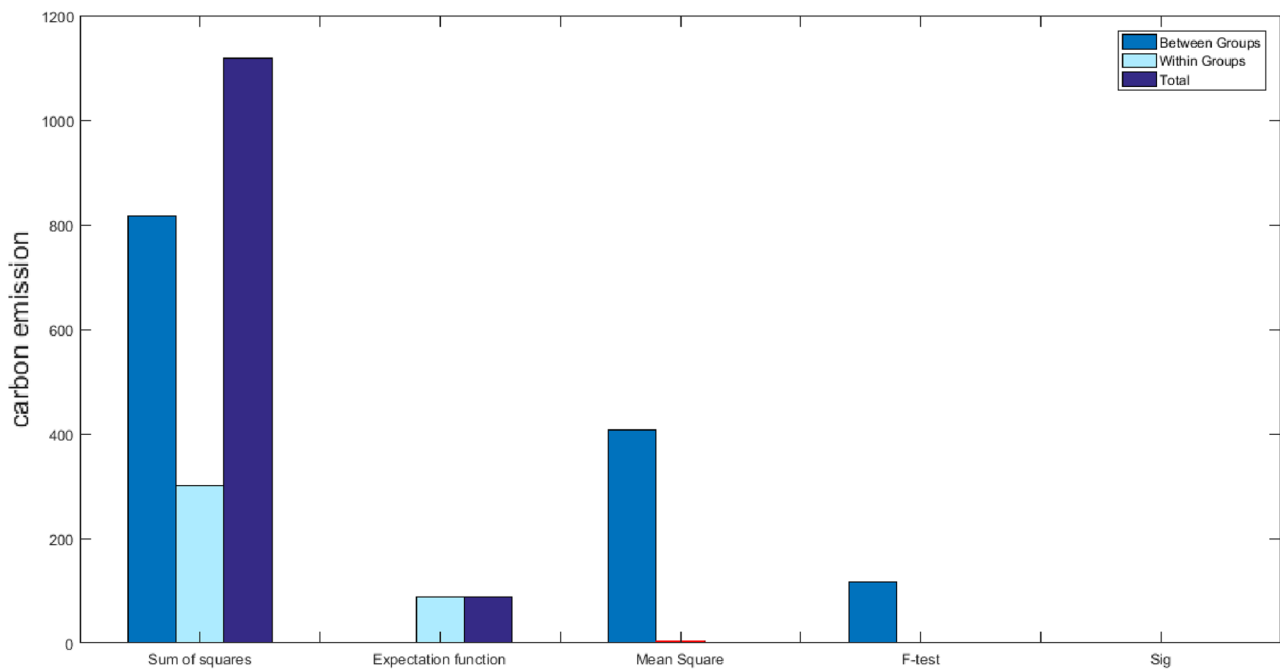


Fig. 7 ANOVA analysis result of carbon emission

solution in the 20th generation and the convergence speed of the proposed IDCQGA is faster than the other four algorithms obviously, moreover, it also has better global optimization capability.

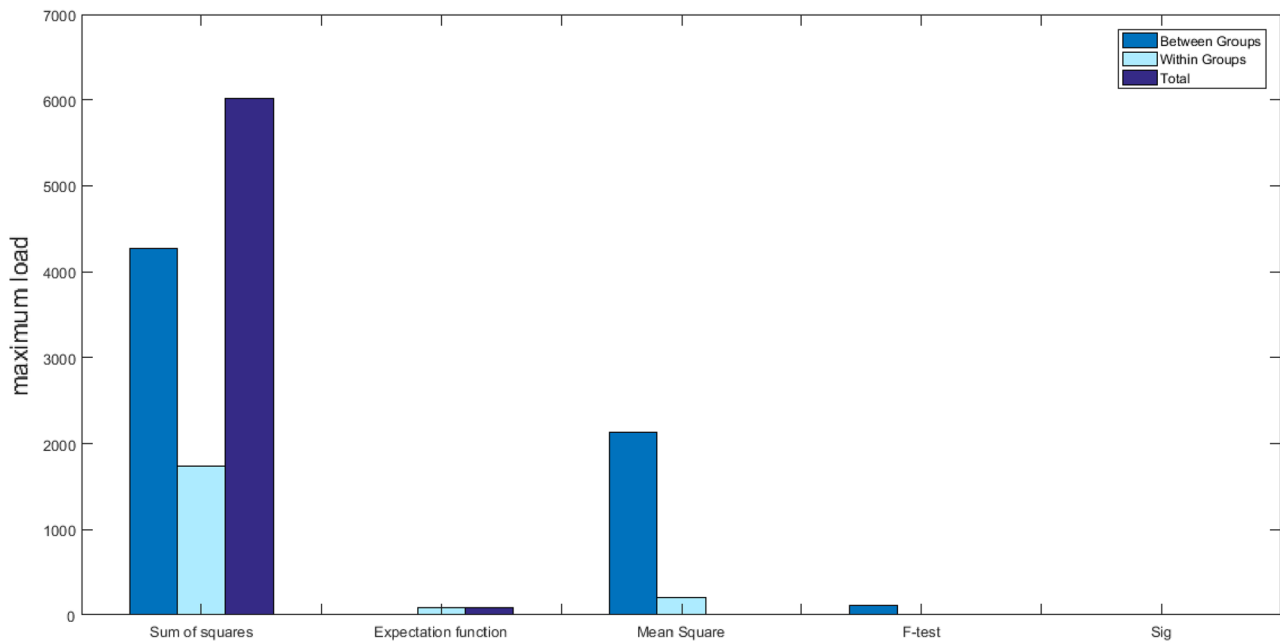


Fig. 8 ANOVA analysis result of maximum load of machine

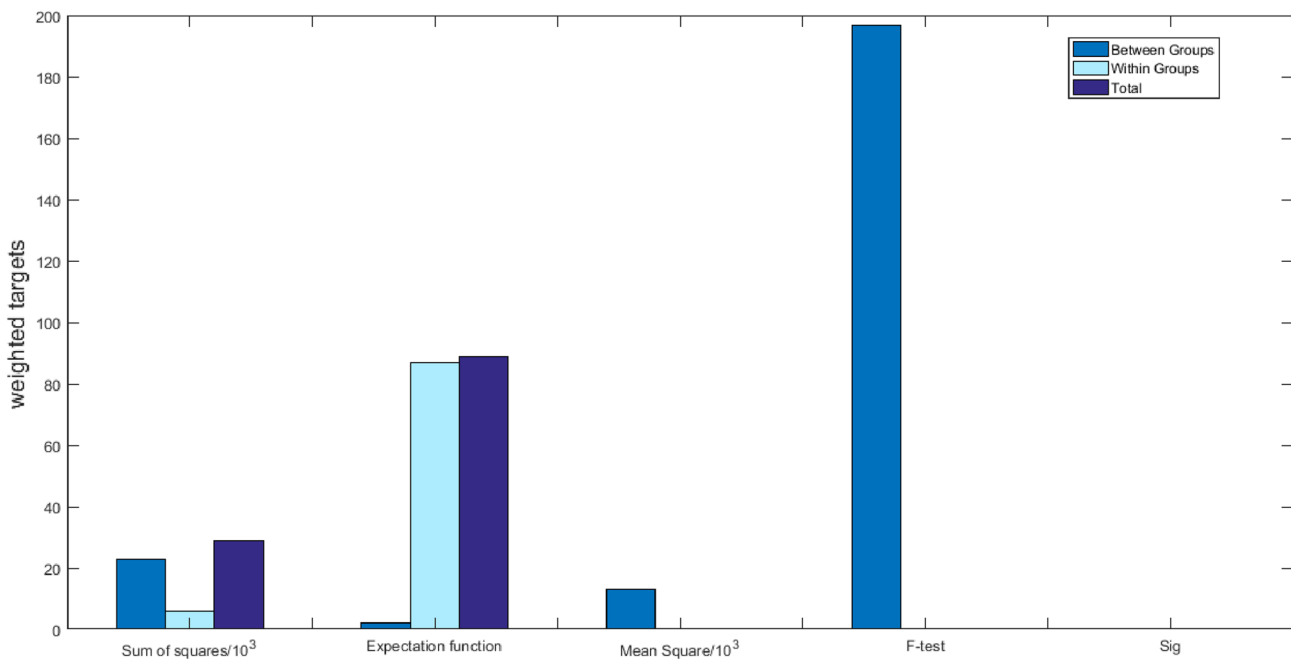


Fig. 9 ANOVA analysis result of weighted targets

5 Conclusion

- (1) An integrated optimization model of shop scheduling with the objective of minimizing carbon emissions and minimizing makespan in manufacturing process is

proposed. According to the feature of multiple optimization parameters in the integrated model, the carbon emission equations, the optimized quantum genetic algorithm, the double-chain coding method and the non-dominant Pareto solution ordering method are designed to sort the population, and obtain good indi-

Table 10 The frontier solution obtained with IDCQGA and HQGA

No.	IDCQGA			HQGA		
	Makespan	Carbon mission	Machine loader	Makespan	Carbon mission	Machine loader
1	286	223.46	235	290	222.58	235
2	289	224.35	226	295	221.55	232
3	290	221.29	226	301	221.24	230
4	290	222.66	235	308	220.24	229
5	294	220.03	230	313	219.47	227
6	295	219.61	228	322	219.01	231
7	296	222.87	227	291	228.08	225
8	297	219.01	226			
9	299	222.74	225			

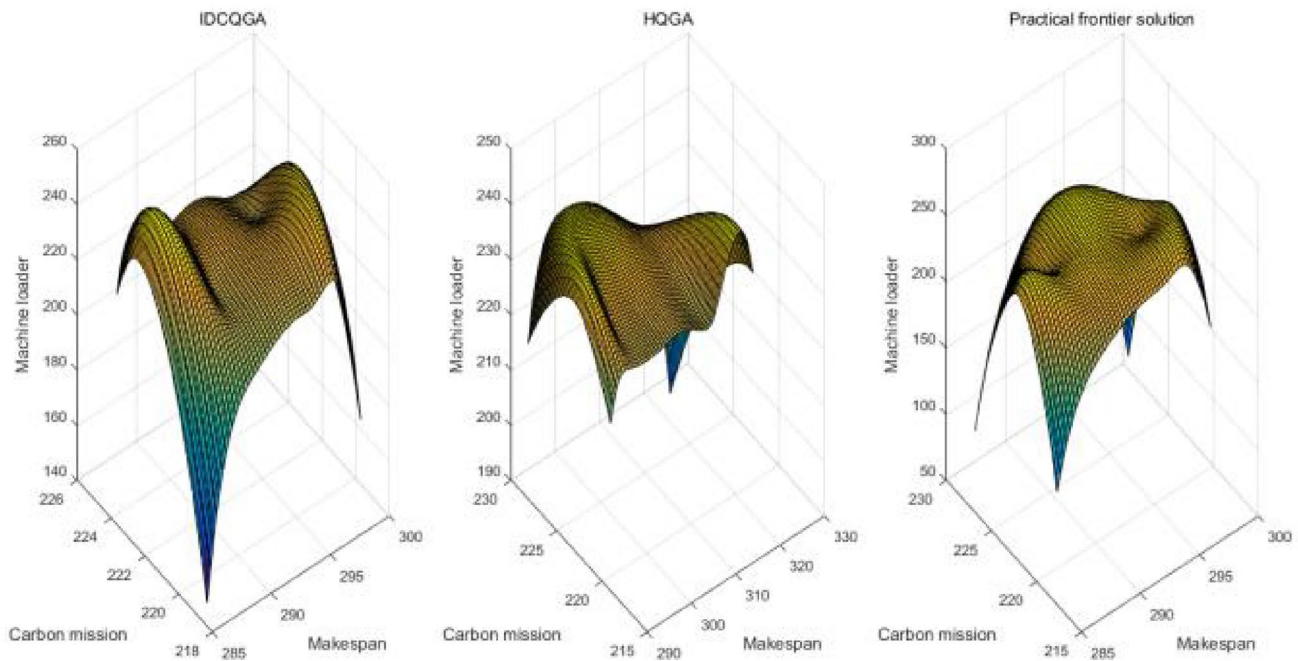


Fig. 10 3D graph of frontier solution obtained by IDCQGA and HQGA

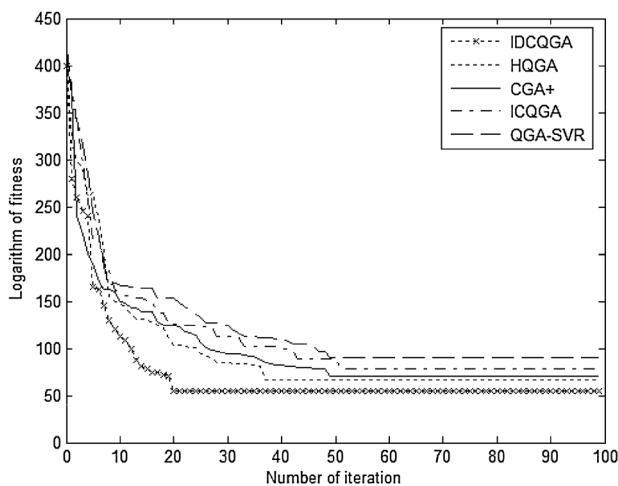
viduals based on the quantum Hadamard gate mutation strategy. Finally, through example verification and comparison with the three commonly used algorithms, each target value is obtained, and then the objective parameters are processed by the variance analysis. It shows that the integrated optimization model and improved quantum genetic algorithms proposed in this paper can effectively solve low-carbon shop scheduling problem.

- (2) The possibility of improved quantum genetic algorithm and shop scheduling integration optimization to further reduce carbon emissions in the manufacturing process is explored. During formulating the scheduling

scheme, only performance indicators such as machine processing time are involved, and other environmental factors such as peak power of machine processing are not considered. When optimizing carbon emission considering machine energy consumption, the change of machine energy consumption power is not considered, and the influence of peak power is ignored. There is still a distance between the research results and practical application, and the research has some limitations. The next step is to design scheduling model with all feasible processing route to improve the practicability of the model.

Table 11 The practical frontier solution obtained with IDCQGA and HQGA

Number	Practical frontier solution		
	Makespan	Carbon emission	Machine loading
1	286	223.46	235
2	289	224.35	226
3	290	221.29	226
4	290	222.66	235
5	291	228.08	225
6	294	220.03	230
7	295	219.61	228
8	296	222.87	227
9	297	219.01	226
10	299	222.74	225

**Fig. 11** The fitness convergence curve of five algorithms

Acknowledgements Foundation items: Project supported by Liaoning Provincial Natural Science Foundation (20180550499, 2019-ZD-0109) and Education Department Project of Liaoning Province (JDL2019022).

Declarations

Conflict of interest None. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Brandimarte P (2011) Decision making under risk. John Wiley & Sons Inc

- Caijie L, Zhitao X, Qin Z et al (2020) Green scheduling of flexible job shops based on NSGA-II under TOU power price. *China Mech Eng* 31(05):576–585
- Deguang L, Taihua Z, Weiping X (2020) Prediction and analysis of workpiece surface roughness based on QGA-SVR. *Mach Too Hydraul* 15:103–108
- Gong X, De Pessemier T, Martens L et al (2019) Energy- and labor-aware flexible job shop scheduling under dynamic electricity pricing: a many-objective optimization investigation. *J Clean Prod* 209:1078–1094
- Jiang TH (2018) Low-carbon workshop scheduling problem based on grey wolf optimization. *Comput Integr Manuf Syst* 24(10):2428–2435
- Jin Z-F, Hou Z-Q, Yu W-S et al (2020) An object tracking approach based on quantum genetic algorithm. *Acta Electron Sin* 11(8):1493–1501
- Kim S, Meng C, Son Y-J (2017) Simulation-based machine shop operations scheduling system for energy cost reduction. *Simul Model Pract Theory* 77:68–83
- Li J-Q, Sang H-Y, Han Y-Y et al (2018) Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *J Clean Prod* 181:584–598
- Ming L, Deming L (2019a) Novel imperialist competitive algorithm for many-objective flexible job shop scheduling. *Control Theory Appl* 36(6):893–901
- Ming L, Deming L (2019b) Research on flexible job shop low carbon scheduling with setup times and key objectives. *J Mech Eng* 21:139–149
- Mishra A, Shrivastava D (2018) A TLBO and a Jaya heuristics for permutation flow shop scheduling to minimize the sum of inventory holding and batch delay costs. *Comput Ind Eng* 124:509–522
- Ning T, Wang XP (2018) Study on disruption management strategy of job-shop scheduling problem based on prospect theory. *J Clean Prod* 194:174–178
- Ning T, Jin H, Song X et al (2018a) An improved quantum genetic algorithm based on MAGTD for dynamic FJSP. *J Ambient Intell Humaniz Comput* 9(4):931–940
- Ning T, Wang XP, Hu XP et al (2018b) Disruption management optimal scheduling for logistics distribution based on prospect theory. *Control Decis* 33(11):2064–2068
- Ning T, Wang XP, Hu XP (2019a) Disruption management decision model for vehicle scheduling under disruption delay. *Syst Eng Theory Pract* 39(5):1236–1245
- Ning T, Wang XP, Hu XP (2019b) Disruption management decision model for vehicle scheduling under disruption delay. *Syst Eng Theory Pract* 39(5):1236–1245
- Ning T, Wang Z, Zhang P et al (2020) Integrated optimization of disruption management and scheduling for reducing carbon emission in manufacturing. *J Clean Prod* 263:1–8
- Piroozfard H, Wong KY, Wong WP (2018) Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. *Resour Conserv Recycl* 128:267–283
- Reddy MS, Ratnam C, Rajyalakshmi G, Manupati VK (2018) An effective hybrid multi objective evolutionary algorithm for solving real time event in flexible job shop scheduling problem. *Measurement* 114:78–90
- Shan G-L, Xu G-G, Qiao C-L (2020) A non-myopic scheduling method of radar sensors for maneuvering target tracking and radiation control. *Def Technol* 16(01):242–250
- Shao X, Liu W, Liu Q, Zhang C (2013) Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem. *Int J Adv Manuf Technol* 67(9–12):2885–2901
- Sheremetov L, Martínez-Muñoz J, Chi-Chim M (2018) Two-stage genetic algorithm for parallel machines scheduling problem:

- cyclic steam stimulation of high viscosity oil reservoirs. *Appl Soft Comput* 64:317–330
- Wang S, Wang X, Yu J et al (2018a) Bi-objective identical parallel machine scheduling to minimize total energy consumption and makespan. *J Clean Prod* 193:424–440
- Wang L, Wang J-J, Wu C-G (2018b) Advances in green shop scheduling and optimization. *Control Decis* 33(3):385–391
- Wu X, Sun Y (2018a) Flexible job shop green scheduling problem with multi-speed machine. *Comput Integr Manuf Syst* 24(04):862–875
- Wu X, Sun Y (2018b) A green scheduling algorithm for flexible job shop with energy-saving measures. *J Clean Prod* 172:3249–3264
- Yang X-L, Hu R, Qian B et al (2019) Enhanced estimation of distribution algorithm for low carbon scheduling of distributed flow shop problem. *Control Theory Appl* 36(05):803–815
- Yansong L, Zongyan W, Ruimin S et al (2020) The improved cloud quantum genetic algorithms for rapidly lightweight of bridge crane main girder. *Mach Des Manuf Eng* 49(1):25–29
- Zeng Z, Hong M, Man Y et al (2018) Multi-object optimization of flexible flow shop scheduling with batch process –consideration total electricity consumption and material wastage. *J Clean Prod* 183:925–939
- Zheng X, Ling W (2018) A knowledge –guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem. *Int J Prod Res* 54(18):1–13
- Zhu H, Deng Q, Zhang L et al (2020) Low carbon flexible job shop scheduling problem considering worker learning using a memetic algorithm. *Optim Eng* 21:1691–1716

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.