ISSN: 2688-9323

Improved Adaptive Genetic Algorithm for Flexible Job Shop Scheduling Problem

Guoliang Li

School of Shanghai Maritime, Shanghai 200000, China

Abstract

In recent years, the energy consumption of manufacturing industry is increasing, resulting in a large number of greenhouse gases. In the context of "carbon neutrality", it is particularly important to study Flexible Job Shop Scheduling Considering carbon emissions. Taking the automobile engine cooling system as the research object, a flexible job shop scheduling model with the optimization objectives of minimum maximum completion time, and minimum carbon emission is established, and an improved genetic algorithm with adaptive adjustment of cross and mutation(IAGA) is designed to solve this model. Adopt double-layer coding for machines and processes. The roulette method and elite replacement strategy are adopted for selection, The roulette method and elite replacement strategy are adopted for selection, and the uniform crossover and pox crossover operations are adopted for machine chromosomes and process chromosomes respectively. The crossover and mutation probabilities are adaptively adjusted according to the individual fitness value, which improves the optimization ability and convergence speed of the algorithm. Finally, the example is simulated and tested by Python software, and the results are compared with those of standard genetic algorithm to prove the feasibility and superiority of the algorithm. At the same time, the energy consumption factor is incorporated into the model, which also effectively reduces the carbon emission, and achieves the effect of combined optimization of completion time and carbon emission. Production enterprises should combine the actual production situation, take the production efficiency into account, and strive to minimize carbon emission and green production.

Keywords

Flexible Job Shop Scheduling; Carbon Emission; Improved Adaptive Genetic Algorithm.

1. Introduction

For a long time, the manufacturing industry is the main body of the national economy, and the production scheduling problem is an important problem in the manufacturing field. A reasonable and efficient scheduling scheme can effectively reduce the production cost of manufacturing enterprises and improve the competitiveness of enterprises. However, the rapid development of manufacturing has also brought serious environmental problems. Nearly one-third of the global energy is consumed by the manufacturing industry, resulting in a large number of greenhouse gases. Figure 1 shows the energy consumption trend of China's manufacturing industry over the years. With the intensification of global competition, controlling the energy consumption during machining to reduce carbon emissions is a powerful way for manufacturing enterprises to improve economic benefits, social benefits and industry competitiveness.

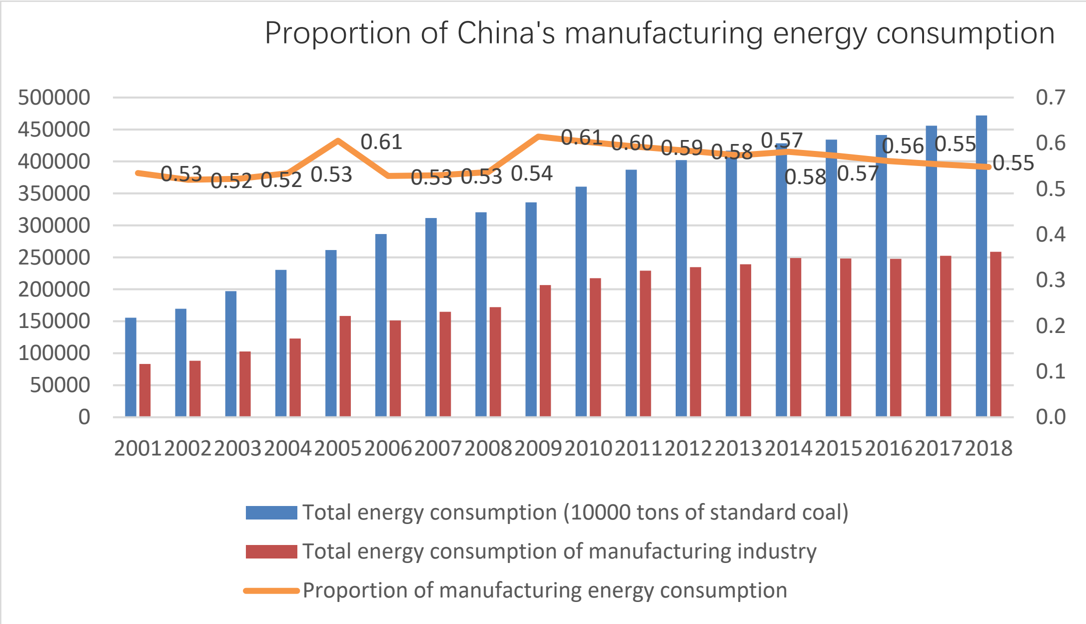

Figure 1. Energy consumption trend of China's manufacturing industry over the years

Production scheduling problem is a typical combined optimization problem. Flexible job shop scheduling problem (FJSP) is an extension of production scheduling problem. The difference is that it allows different processes of the same job to be processed on different machines, which is more appropriate to the actual processing situation of modern job shop. Flexible job shop scheduling problem has the characteristics of complexity, dynamics and multi constraints. It is a typical NP hard problem. The single objective flexible job shop scheduling problem has been thoroughly studied. Minimizing the maximum completion time has always been the focus of previous scholars.For this optimization goal, Jiang Tianhua[1]proposed a hybrid gray wolf optimization algorithm (HGWO). Zhang Tongrui [2] proposed a hybrid competitive group optimization algorithm to solve it. Zheng Jie [3] proposed a hybrid competitive group optimization algorithm to solve it. Zheng Jie [4] proposed immune genetic algorithm. Chen Jinguang [5] designed an improved genetic algorithm for job shop scheduling. Gong [6] established a scheduling model considering worker activity and solved it by hybrid artificial bee colony algorithm (HABCA). Zhang[7]proposed an improved memetic algorithm to solve the flexible job shop scheduling problem with transportation time.Li[8]mixed genetic algorithm (GA) and tabu search (TS) to solve the flexible job shop scheduling problem. Zhang Guijun[9]proposed a dynamic strategy differential evolution algorithm, which dynamically selects the mutation strategy according to the congestion degree between individuals, designs the algorithm and obtains the optimal scheduling scheme. Zhou Yanping[10]and others designed an adaptive differential evolution hybrid algorithm and applied it to the field of production scheduling.

Many scholars have also studied the multi-objective flexible job shop scheduling problem. Zhang[11]proposed an improved genetic algorithm to solve the production scheduling problem of flexible job shop including processing time, preparation time and transportation time. J.S. Sadaghiani[12]and others built a job shop scheduling model with the goal of reducing processing span, total workload and maximum load, and used a comprehensive heuristic algorithm to solve it. Luo[13]proposed the distributed flexible job shop scheduling problem (DFJSPT) and established a scheduling model with the shortest completion time and the lowest total energy consumption as the optimization objectives. Wang Yan[14]and others proposed an improved multi-objective differential evolution algorithm to solve the multi-objective dynamic flexible job shop scheduling problem. Aiming at the scheduling problem in flexible manufacturing system (FMS), Zhong Zhiqing[15]studied how to reduce carbon dioxide emissions from machine tool standby and processing on the basis of reducing the completion time. Huang[16]proposed a hybrid genetic particle swarm optimization algorithm based on teaching and learning to solve the multi-objective flexible job shop scheduling problem. Yin[17]proposed a multi-objective genetic algorithm based on simplex lattice design to solve the low-carbon mathematical scheduling model for the flexible workshop environment. Considering the transportation time of workpieces, Li Xiangyi[18]established a green flexible job shop scheduling model with the optimization objectives of completion time, carbon emission and machine load, and designed an improved particle swarm optimization algorithm to solve it. Dong Hai[19]combined machine flexibility, worker flexibility and parallel process flexibility, established a multi flexible job shop scheduling model and proposed a multi-objective optimization algorithm.

At present, scholars at home and abroad have done a lot of research on the single objective and multi-objective production scheduling problem of flexible job shop, and applied and developed various algorithms, but there are still deficiencies. For example, few scholars have studied the energy conservation and emission reduction of production and manufacturing. Under the new environmental protection concept of "carbon harmony", production enterprises must implement energy conservation and emission reduction strategies, Otherwise, it will gradually lose its market competitiveness under such a general trend. This paper will comprehensively consider the objectives pursued by enterprises, introduce carbon emissions into the research of production scheduling, and minimize carbon emissions on the basis of shortening the maximum completion time.

2. Problem Description

Job shop scheduling problem is a typical NP hard problem. Compared with the traditional job shop scheduling, it breaks through the limitation of pre specifying the process route of processing jobs, increases the machine flexibility, and is conducive to the improvement of job shop production efficiency and machine utilization. FJSP can be described as: there are n independent jobs to be processed, which need to be processed on m machines, h_i is the total number of processes of the i-th job, O_{ij} is the j-th process of the i-th job. Each job has at least one process, and each process has at least one Machinable machine. The processing time of different machines is different. The goal of production scheduling is to make each process have appropriate machines to process under the condition of limited resources, and determine the starting processing time and processing sequence of the process, so as to obtain the optimal scheduling scheme and optimize the given performance index.

3. Mathematical Model of FJSP

3.1. Parameter Definition

The mathematical model of FJSP is relatively complex. In order to facilitate subsequent description, the following contents shall be defined before establishing the model:

N : Set of jobs, $N = \{1, 2...i...n\}$.

K : Set of machines, $K = \{1, 2...k...m\}$.

H_i: Operation set of the i-th job, $H_i = \{1, 2, ..., h_i\}$.

O_{ij} : The j-th process of the i-th job.

P_{ijk} : Processing time on machine k of O_{ij}.

S_{ij} : Start processing time of O_{ij} .

C_{ij} : End processing time of O_{ij}.

C_i : Completion time of the i-th job.

$C_{\max}$: The maximum completion time, that is, the time for all jobs to complete processing.

E_k : Rated power during machine k processing.

EF : Carbon emission coefficient of electric power.

B : A big number.

3.2. Decision Variables

x_{ijk} : If machine k is selected for the operation O_{ij}, it is 1, otherwise it is 0.

y_{ijhlk} : It is 1 if the operation O_{ij} is processed on the machine k before O_{hl}, otherwise, it is 0.

3.3. Objective Function

$$\min C_{\max} = \max_{1\le i\le n}(C_i). \tag{1}$$

$$\min E_T = \sum_{i=1}^{n}\sum_{j=1}^{hi}\sum_{k=1}^{m} P_{ijk}x_{ijk}E_k EF. \tag{2}$$

Equations (1-2) are the optimization objectives, equation (1) represents the minimization of the maximum completion time, and equation (2) represents the minimization of carbon emission.

The FJSP model needs to meet the following constraints:

$$S_{ij} + \sum_{k=1}^{m} P_{ijk}\cdot x_{ijk} = C_{ij}, \quad \forall i \in N, j \in H_i. \tag{3}$$

Equation (3) indicates that once a process of the job starts processing, it cannot be interrupted until the processing is completed.

$$C_{ij} \le S_{i(j+1)}, \quad \forall i \in N, \ j, j+1 \in H_i. \tag{4}$$

Equation (4) indicates that the processing sequence of the same job must be carried out according to the process sequence.

$$C_i \le C_{\max}, \quad \forall i \in N. \tag{5}$$

Equation (5) indicates that the completion time of each job shall not exceed the completion time of all jobs.

$$S_{hl} + B\left(1-y_{ijhlk}\right), \quad \forall i, \ h \in N, j, l \in H_i, k \in K. \tag{6}$$

Equation (6) indicates that any machine at any time is only allowed to process one process at the same time.

$$\sum_{k=1}^{m} x_{ijk} = 1, \forall i \in N, j \in H_i. \tag{7}$$

Equation (7) indicates that only one machine can be selected for each process.

$$x_{ijk} + x_{hlk} \geq 2y_{ijhlk}, \forall i, h \in N, j, l \in H_i, k \in K. \tag{8}$$

Equation (8) indicates that each machine may process more than one process.

$$S_{ij} \geq 0, C_{ij} \geq 0, \forall i \in N, j \in H_i. \tag{9}$$

Equation (9) indicates that the start processing time and completion time of any process are non negative, and any workpiece can be processed from time 0.

3.4. Method for Solving Double-objective

Because the three objectives of maximum completion time and carbon emission are inconsistent, there is no scheduling scheme to optimize the double objectives at the same time. Therefore, this paper adopts the weighted method to convert it into a single objective for solution.

$$F(x) = v_1 \cdot C(x) + v_2 \cdot E(\mathrm{x}). \tag{10}$$

In equation (10), V1 and V2 represent the weight coefficients of each target, and the sum of them is equal to 1. When dealing with the Double-objective optimization problem, the decision-maker should set a reasonable weight for each objective in combination with the actual production situation and the importance of each objective.

4. Improved Adaptive Genetic Algorithm to Solve Double-objective FJSP Model

Genetic algorithm is widely used in the field of production scheduling because of its strong robustness. However, previous studies also reflect the problem that genetic algorithm is easy to precocious. In this paper, genetic algorithm is improved, roulette and elite replacement strategy are adopted for selection, and adaptive cross mutation probability is adopted, which can effectively enhance the ability of population evolution and avoid falling into local optimization and premature.

4.1. Chromosome Coding

This paper adopts a double-layer coding method based on process sequence (OS) and machine allocation (MS). The coding based on the process processing sequence, i.e. the processing sequence of the process, refers to the chromosome formed by the process processing sequence of all jobs. The coding based on machine allocation, i.e. processing machine sequence, refers to the chromosome formed by the machine selected during processing in the corresponding process. The length of the two chromosomes is the same, and the integer coding method is adopted. The specific coding method is shown in Figure 2.

Operation coding	2	1	3	2	1	3	1	2
Machine coding	4	2	3	1	2	1	3	1

Figure 2. Coding diagram

The length of the process code is equal to the total number of all processes. The number i of the operation coding layer represents the i-th job, and the number of occurrences j represents the j-th process of the job. For example, "2" in the first position of the operation coding layer indicates the operation O_{21}, "1" in the second position indicates the operation O_{11}, and "2" in the fourth position indicates the operation O_{22}. The length of the machine code is equal to the length of the operation code, which corresponds to the processing machine of each operation in the operation code layer. "4" in the first position of the machine coding layer indicates that the operation O_{21} is processed on the machine 4, "2" in the second position indicates that the the operation O_{11} is processed on the machine 2, and "1" in the fourth position indicates that the operation O_{21} is processed on the machine 1.

4.2. Chromosome Decoding

Decoding is to restore the chromosome into a scheduling scheme in combination with the job shop scheduling problem. This paper adopts plug-in greedy decoding, that is, the unscheduled process is inserted into the earliest feasible processing time on the corresponding idle machine according to its processing time without changing the start time of other scheduled processes. The specific steps are as follows: first, obtain the machine selection scheme of each process through machine vector coding, then determine the processing time of the process in the corresponding processing machine, finally determine the processing sequence of the process according to the process vector and decode it in turn. Without delaying the start time of other scheduled processes, Insert the process into the earliest feasible processing time on the corresponding machine for processing.

4.3. Fitness Calculation

Since the flexible job shop scheduling problem is a multi-objective combinatorial optimization problem, the fitness function should be considered comprehensively in combination with completion time, machine load and carbon emission. Assuming that the scheduling plan of a flexible job shop, C (v) and E (v) represent the maximum completion time and carbon emission generated by processing completion corresponding to the v-th individual respectively, the fitness function of the problem can be expressed as:

$$f(v) = w_1 \cdot \frac{C_{max} - C(v)}{C_{max} - C_{min}} + +w_2 \cdot \frac{E_{max} - E(v)}{E_{max} - E_{min}}. \tag{11}$$

Where w1, w2 represent the weight coefficient of each index, and the sum of the three is equal to 1. C_{max} and C_{min} respectively represent the maximum and minimum values of C (v) in the population, and others are the same.

4.4. Genetic Operation

4.4.1. Select Operation

At the same time, in order to obtain better individuals and improve the calculation speed of the algorithm, this paper adopts the elite replacement strategy instead of the elite retention strategy. Through each iteration of the algorithm, the first 10% individuals with high fitness value in the population are retained to replace the last 10% individuals with low fitness value, so as to eliminate the poor individuals and speed up the population convergence.

4.4.2. Crossing Operation

In this paper, two different crossover methods will be adopted to cross the machine chromosome and process chromosome.

(1) Machine distribution section

The machine allocation part must ensure that the sequence of each gene remains unchanged and adopt uniform crossover operation.

Step 1: Randomly generate an integer r in the interval.

Step 2: Randomly generate r unequal integers according to the random number r.

Step 3: According to the integer r generated in step 2, copy the genes at the corresponding positions in the parent chromosomes P1 and P2 into the offspring chromosomes C1 and C2, and maintain their position and order.

Step 4: Copy the remaining genes of P1 into C2 in the original order. Similarly, copy the remaining genes of P2 into C1 in the original order and maintain their position and order.

Two positions are determined according to the random number, the genes with the blue part of chromosomes P1 and P2 are sequentially copied into offspring chromosomes C1 and C2, and then the remaining genes of P1 and P2 are sequentially copied into C2 and C1, resulting in a new individual, see Figure 3.

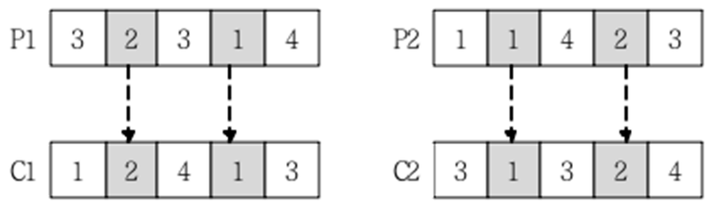

Figure 3. Uniform crossing

(2) Operation sequencing

The process chromosome adopts the crossing method based on job sequence (POX), and the crossing process is as follows:

Step 1: randomly select two chromosomes in the population as parent chromosomes P1 and P2, i.e., parent1 and parent2.

Step 2: Randomly divide the job set into two non-empty job sets G1 and G2.

Step 3: transfer the job containing the job set G1 in the parent chromosomes P1 and P2 to the offspring chromosomes C1 and C2 according to the original position.

Step 4: put the artifacts containing the artifact set G2 in the parent chromosomes P1 and P2 into the offspring chromosomes C2 and C1 in order, and ensure that their positions remain unchanged.

As shown in Figure 4, there are four jobs [1,2,3,4] to be processed, which are divided into two jobs subsets [2] and [1,3,4], then copy the part containing jobs set G1 in P1 (blue part) into C1, and copy the gene containing G1 in P2 (blue part) into C2, keeping the original position unchanged. Finally, the gene excluding G1 in P1 is copied to C2 in order, and the gene without G1 in P2 is copied to C1, and the new individual is obtained by crossing.

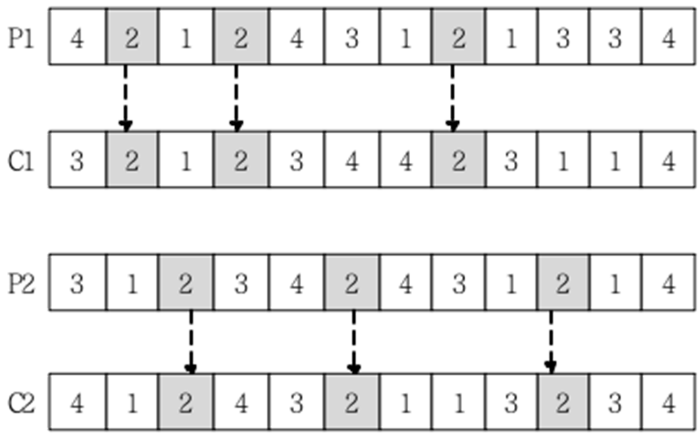

Figure 4. Pox crossover

4.4.3. Mutation Operation

The mutation operator in this paper includes two parts: machine chromosome mutation and process chromosome mutation. In order to ensure that the newly obtained chromosome is schedulable, for the process coding layer, when the chromosome changes, the machine coding corresponding to the variation process remains unchanged. In order to ensure the effectiveness of chromosomes, two-point exchange mutation is adopted for process chromosomes. For the machine coding layer, a position is randomly selected for variation, and the variation method is to select a machine replacement from the alternative processing machine of the process.

4.4.4. Adaptive Crossover Mutation Probability

The traditional genetic algorithm usually sets a fixed value for the crossover probability and mutation probability. However, the value given according to previous experience or experimental results has great subjective randomness. When the probability of crossover mutation is set to be large, the existing excellent individuals may become worse after a series of crossover mutation operations, and the determination is too small, which is not conducive to searching for a better solution, Make the algorithm fall into the problem of premature. This paper adopts the solution of adaptive adjustment of crossover and mutation probability [5], which can effectively enhance the ability of population evolution and avoid falling into local optimization and precocity. The adaptive adjustment crossover probability formula and adaptive adjustment mutation probability formula are shown in formula (12) and formula (13) respectively.

$$P_c = \left\{ \begin{array}{ll} k_1 - \dfrac{k_1(f' - f_{avg})}{f_{\max} - f_{avg}}, & f' > f_{avg} \\ k_2, & f' \le f_{avg} \end{array} \right\}. \tag{12}$$

$$P_m = \left\{ \begin{array}{ll} k_3 - \dfrac{k_3(f - f_{avg})}{f_{\max} - f_{avg}}, & f > f_{avg} \\ k_4, & g \le g_{avg} \end{array} \right\}. \tag{13}$$

ISSN: 2688-9323

In the above formula, $f_{\max}$ represents the maximum fitness value of all chromosomes in the current population, f_{avg} refers to the average fitness value, f' represents the individual with the larger fitness value among the two chromosomes performing crossover operation, and f represents the fitness value of the selected chromosome to perform mutation operation. Among them, k1, k2, k3 and k4 are all within the range of (0, 1).

5. Application and Analysis of Numerical Examples

In order to verify the feasibility of the model established in this paper and the improved adaptive genetic algorithm, an 8 × 8 to verify the feasibility of this algorithm in the application of flexible job shop scheduling. The original data of the example is from the literature [3].

5.1. Parameter Setting

In this paper, Python is used to program and solve the algorithm, and the relevant parameters are set as follows: the population size is set to n = 300, the maximum number of iterations is I = 50, and the crossover probability and mutation probability are adjusted adaptively, which are determined by equations (12) and (13) respectively, where k1 = k2 = 0.9, k3 = k4 = 0.1.

5.2. Algorithm Performance Analysis

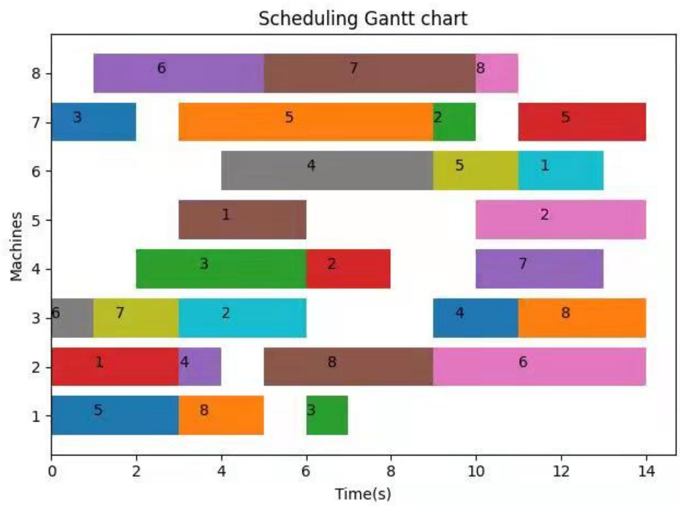

Figure 5. 8 × 8 FJSP scheduling Gantt chart

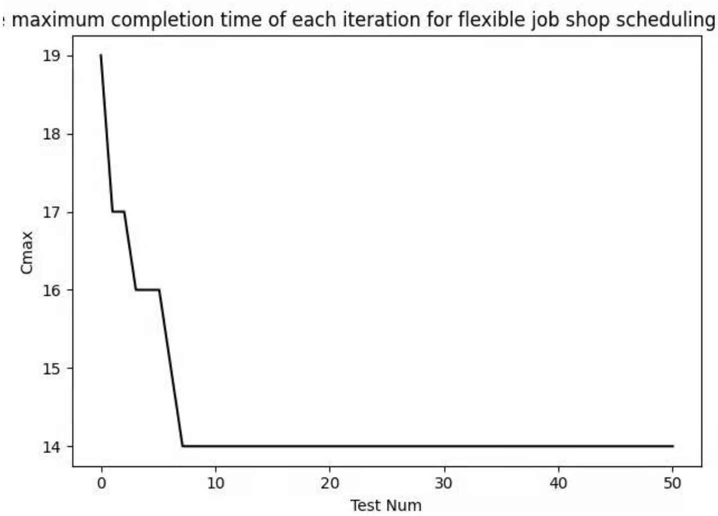

Figure 6. 8 × 8 FJSP convergence curve

ISSN: 2688-9323

For the standard 8×8 example in reference [3], the optimal solution is obtained by using CPLEX, that is, the maximum completion time is 14.

Using Python to program the IAGA algorithm in this paper with 8×8 standard example. The scheduling Gantt chart and convergence curve of 8×8 standard example is obtained as follows, see Figure 5 and Figure 6.

Through the scheduling Gantt chart in Figure 5, we can not only know the maximum completion time of the standard example is 14, which is consistent with the solution results of CPLEX, which verifies the effectiveness of the algorithm in this paper. It can also see the processing machine selected by each process, the transfer trajectory of each job and the completion time of each job through the scheduling Gantt chart. Then, according to formula (2), assuming that each unit (kwh) of electricity consumed will produce 0.96kg of carbon dioxide, that is, EF = 0.96kg/kwh, the carbon dioxide emission at this time is 361.44.

Record 8 ×8 example, runs independently for 20 times, and the analysis and comparison with the solution results of standard genetic algorithm are shown in Table 1.

Table 1. Comparison of solution results between standard GA and IAGA

	Optimal solution (C_{max})	Number of optimal solutions	average value	Average number of iterations	Average convergence rate (%)
IAGA	14	14	14.3	15.6	70
GA	14	6	15.1	27.66	30

As can be seen from table 1, during the simulation of 8×8 examples, the two algorithms have found the optimal solution, while the algorithm in this paper runs independently for 20 times, the number of times of the optimal solution is more, and the mean value is closer to the optimal solution, which shows the effectiveness and stability of the improved genetic algorithm in this paper, and also shows that the improved genetic algorithm in this paper has strong optimization ability, It effectively alleviates the problem of falling into prematurity in standard genetic algorithm, which is also the result of the adaptive adjustment of crossover and mutation probability used in this paper.

5.3. Analysis of Example Results

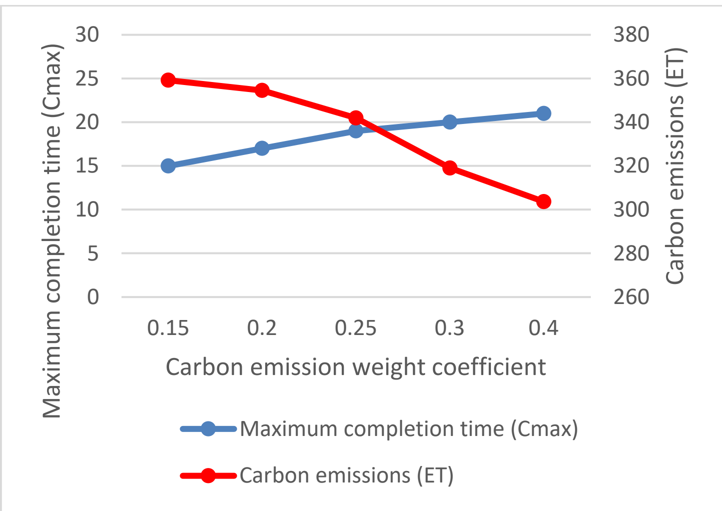

Figure 7. Trend chart of C_{max} and E_T under different weights

In order to study the change of the target value when considering carbon emission and not considering carbon emission, the corresponding weight is set for the target function in combination with relevant literature. For the standard 8×8 example in reference [3], set different proportions for completion time and carbon emission, record the changes of the two target values, and draw the changes of the two target values under different weights, see Figure 7.

Figure 7 shows that the weight of carbon emission increases gradually under different completion time, which may also lead to the gradual decrease of the weight of carbon emission coefficient, Taking into account carbon emissions may reduce the operation efficiency. By adjusting the weight coefficient of the two optimization indicators, the size of the two can be controlled to meet the production demand. Production enterprises should set reasonable weight coefficients in combination with actual production conditions and national energy conservation and emission reduction policies, give overall consideration to production efficiency and energy consumption, and strive to reduce carbon dioxide emissions in the processing process while ensuring production efficiency.

6. Conclusion

In this paper, an improved genetic algorithm with adaptive adjustment of cross mutation is designed to solve the dual objective flexible job shop scheduling problem. Through the CPLEX solution and python simulation of the established dual objective flexible job shop scheduling model, the results show that the improved adaptive genetic algorithm has obvious effectiveness and superiority. Compared with the standard genetic algorithm, it has stronger optimization ability and faster convergence speed, effectively avoids the problem of falling into prematurity, and shows better optimization performance. Through the comparison of the target values under different weights, it shows that the minimization of completion time and energy consumption are two goals that conflict with each other. Pursuing the maximization of one of them will inevitably worsen the other index. When making production plans, decision makers should set reasonable weights in combination with the actual production situation. However, considering the more complex environment in actual production, no-load machine and power consumption of auxiliary equipment are important factors for carbon emission. The next research will consider the energy conservation and emission reduction in flexible job shop scheduling more comprehensively and carefully, and continuously improve the quality of the solution to make it more meaningful.

References

[1] Zheng Jie, pan Dazhi An improved discrete firefly optimization algorithm for flexible job shop scheduling [J] Computer and modernization, 2021 (08): 11-15 + 23 + 29.

[2] Zhang Tongrui, Wu Dinghui Hybrid competitive group optimization algorithm for flexible job shop scheduling [J] Control engineering, 2021,28 (09): 1820-1828.

[3] Jiang Tianhua Hybrid grey wolf optimization algorithm for flexible job shop scheduling [J] Control and decision making, 2018,33 (03): 503-508.

[4] Cao Kunyu, Chen yongdang, song Xinxin Improved immune genetic algorithm for flexible job shop scheduling problem [J] Computer technology and development, 2020, 30 (11): 174 – 179.

[5] Chen Jinguang, Ma Lingye, Ma Lili Improved genetic algorithm for job shop scheduling problem [J] Computer system application, 2021, 30 (05): 190 – 195.

[6] GONG G, CHIONG R, DENG Q. A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility[J]. International Journal of Production Research, 2020, 58(14): 4406–4420.

[7] ZHANG G, SUN J, LU X. An improved memetic algorithm for the flexible job shop scheduling problem with transportation times[J]. Measurement and Control, 2020, 53(7–8): 1518–1528.

[8] LI X, GAO L. An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem[J]. International Journal of Production Economics, 2016, 174: 93–110.

[9] Zhang Guijun, Ding Qing, Wang Liujing, Zhou Xiaogen Optimization method of production scheduling in flexible Workshop [J] Computer science, 2018, 45 (02): 269 – 275.

[10] Zhou Yanping, Cai Su, Li Jinpeng A hybrid algorithm of particle swarm optimization and improved adaptive differential evolution and its application in production scheduling [J] Computer measurement and control, 2019, 27 (08): 227 – 230.

[11] ZHANG G, HU Y, SUN J. An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints[J]. Swarm and Evolutionary Computation, 2020, 54: 100664.

[12] SADAGHIANI J S, BOROUJERDI S A, MIRHABIBI M. A Pareto archive floating search procedure for solving multi-objective flexible job shop scheduling problem[J]. Decision Science Letters, 2014, 3(2): 157–168.

[13] LUO Q, DENG Q, GONG G. An efficient memetic algorithm for distributed flexible job shop scheduling problem with transfers[J]. Expert Systems with Applications, 2020, 160: 113721.

[14] Wang Yan, Ding Yu Optimal scheduling and decision-making method of dynamic flexible job shop [J] Journal of system simulation, 2020, 32 (11): 2073 – 2083.

[15] Zhong Zhiqing Research on low carbon production scheduling method for FMS [J] Manufacturing technology and machine tools, 2019 (07): 23 – 27.

[16] HUANG X, GUAN Z, YANG L. An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem[J]. Advances in Mechanical Engineering, 2018, 10(9): 168781401880144.

[17] YIN L, LI X, GAO L. A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem[J]. Sustainable Computing: Informatics and Systems, 2017, 13: 15–30.

[18] Li Xiangyi, he Xingyue, Wang Lei Research on green Flexible Job Shop Scheduling Considering workpiece transportation time [J] Digital manufacturing science, 2020, 18 (02): 102 – 106.

[19] Dong Hai, Xu Xiaopeng, thank you Multi objective optimization algorithm for solving multi flexible job shop scheduling problem [J] Computer science, 2020, 47 (12): 239 – 244.