

Received June 21, 2021, accepted July 18, 2021, date of publication July 21, 2021, date of current version August 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098823

# Improved Adaptive Non-Dominated Sorting Genetic Algorithm With Elite Strategy for Solving Multi-Objective Flexible Job-Shop Scheduling Problem

XU LIANG<sup>1,2</sup>, JIABAO CHEN<sup>2</sup>, XIAOLIN GU<sup>1,2</sup>, AND MING HUANG<sup>1,2</sup>

<sup>1</sup>Computer School, Beijing Information Science and Technology University, Beijing 100192, China

<sup>2</sup>Software Technology Institute, Dalian Jiaotong University, Dalian 116021, China

Corresponding author: Xiaolin Gu (guxiaolin60@126.com)

This work was supported by the Liaoning National Natural Science Foundation of China under Grant 20170540131.

**ABSTRACT** Regarding the complicated flexible job-shop scheduling problem, it is not only required to get optimal solution of the problem but also required to ensure low-carbon and environmental protection. Based on the NSGA-II algorithm, this article proposes an improved adaptive non-dominated sorting genetic algorithm with elite strategy (IA-NSGA-ES). Firstly, the constructive heuristic algorithm is introduced in the initial population phase, and the weight aggregation method is used to restrain the multi-objective mathematical model which takes total completion time, carbon emission and maximum machine tools load as objectives; secondly, elite strategy is improved, simulated annealing method is used to replace parent generation by child generation to enhance the replaced population quality. The improved algorithm obtains the Pareto optimal solution set faster. Using standard computation example and practical workshop problem for simulation, the results of simulation prove that the algorithm is effective and feasible.

**INDEX TERMS** Non-dominated sorting, genetic algorithm, adaptive, job-shop scheduling.

## I. INTRODUCTION

With the continuous progress of the national economy, the intelligent manufacturing industry has been fully developed. As the core of the manufacturing production, production workshop scheduling plays an important role in improving the production efficiency of enterprises and saving production costs [1]. The traditional workshop scheduling method can no longer meet the needs of the manufacturing industry. Flexible job-shop scheduling problem is the extension of the traditional job-shop scheduling problem raised firstly by Bruker and Schlies for the first time in 1990 [2]. Relative to JSP problem, FJSP allows different jobs of multiple operations to be machined on different machines, changing the uniqueness restriction of equipment, and it is possible to select machine according to load conditions such as machine resources, increasing the flexibility of processing, this is more in line with the actual production situation of enterprises. In the research of shop scheduling optimization

problem, the Multi-Objective Flexible Job-shop Scheduling Problem (MOFJSP) is the most difficult and closest to the actual production environment of the shop scheduling optimization problem.

In recent years, scholars at home and abroad have made certain achievements in exploring method for solving FJSP problem. Zi'an *et al.* [3] proposed a double-population hybrid genetic algorithm for solving the flexible job-shop scheduling problem, in which one population focuses on global search, and the other population is mainly responsible for local search, improving the accuracy of algorithm and convergence speed. Xiaotao and Chong [4] proposed a hybrid multi-intelligence genetic algorithm for solving the problem of job-shop scheduling, and made improvement to a certain extent in terms of non-deterministic polynomial characteristics of job-shop scheduling problem and properties of large spatial distribution of solution. For flexible job-shop scheduling of different varieties and small batches, Chen and Li [5] proposed a GA-based heuristic algorithm to separate operation code from machine tools code and respectively perform crossover and mutation operation to speed up convergence

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Pilato<sup>1</sup>.

speed; Cinar *et al.* [6] proposed a priority-based GA for solving flexible job-shop scheduling problem, applying iteration local search (ILS) to chromosome upon end of each reproduction process, and enhancing the performance of genetic algorithm applied in flexible job-shop scheduling.

In the actual production process, production cost, processing time and low-carbon environmental protection are problem to be considered, it is difficult to reflect the real situations of scheduling of workshop by one objective, at present, many scholars at home and abroad also shift the direction of research to solving multi-objective flexible job-shop scheduling problem. Yan and Wu [7] proposed a super-heuristic genetic algorithm taking minimization of the maximum completion time as the objective function for solving the job-shop scheduling problem. Guohui and Shijie [8] established a low-carbon flexible job-shop scheduling problem model that takes machine speed into account with consideration of double-objective problem of low carbon and completion time, and reduced carbon emission by adjusting non-critical operations after solving the problem targeting completion time by adopting with improved genetic algorithm. Zhang and Chiong [9] established a mathematical model which takes total weighted tardiness and total energy consumption as double-objective with consideration of energy-saving problem, taking energy consumption and carbon emission into account, and proposed a hybrid genetic algorithm with local search strategy for solving the problem. Abreu *et al.* [10] proposed a hybrid genetic algorithm regarding open shop scheduling problem (OSSP), which takes sequence-related establishment time and total completion time as the objective function. Two novel, constructive and heuristic methods are used in the article to generate the initial population. Salido *et al.* [11] studied speed-scaled job-shop scheduling problem, established a mathematical model taking completion time and energy consumption as objective, and proposed genetic algorithm for solving the problem; considering the multi-objective problem, taking minimized completion time, minimum total machine load and minimum critical machine load as objectives, Xiaobing and Yizhen [12] proposed improved genetic algorithm based on tripartite game theory for solving multi-objective flexible job-shop scheduling model; Piroozfard *et al.* [13] proposed an improved multi-objective genetic algorithm to obtain a high-quality non-dominated schedule, its objective is to minimize total carbon footprint and the overall post-work criteria, and respectively act as sustainable and classical objective function.

In 2002, Indian scientist Deb *et al.* [14] proposed a non-dominated sorting genetic algorithm with elite strategies (NSGA-II). NSGA-II algorithm is one of the most widely used and influential multi-objective genetic algorithm. Multi-objective genetic algorithm is an evolutionary algorithm designed to coordinate the relationship between objective functions, and look for the optimal solution set so that each objective function can reach the function value that meets the conditions as much as possible. NSGA-II becomes

one of the basic algorithms for solving multi-objective optimization problem by leveraging advantages such as simplicity and effectiveness. In view of the characteristics of flexible job-shop scheduling, this article designs an improved adaptive non-dominated sorting genetic algorithm with elite strategy (IA-NSGA-ES), and establishes a multi-objective mathematical model of completion time, carbon emission and machine tools maximum load, coding scheme should choose two-layer coding method, machine decoding of decoding program not only considers the processing time of machine but also considers the carbon emission of machine. Improvements mainly include:

- 1) Initial population uses partial construction heuristic algorithm to improve the quality of the generated population;
- 2) Improved adaptive crossover and mutation algorithm is used to speed up convergence speed and prevent local optimization;
- 3) The introduction of the simulated annealing elite strategy, and the shift from the original simple combination of parent population and child population to the simulated annealing screening of the next generation of populations, which are beneficial for enhancement of the quality of the next generation of individuals;
- 4) Improved fast non-dominated sorting method is used to get the Pareto optimal solution. This not only ensures the diversity of population, but also maintains more high-quality individuals, and improves the quality of population.

## II. DESCRIPTION OF PROBLEM

### A. DESCRIPTION OF FJSP PROBLEM

Flexible job-shop scheduling problem is generally described as,  $n$  jobs are processed on  $u$  units of machine; each unit of machine has different gears, different gears correspond to different speeds, each gear uses different powers, and the carbon emission produced are also different; there are totally  $O_i$  operations, each operation can be performed on one or multiple units of machine, it is possible to use all gears of machine, and processing time is not necessarily identical when different gears are used. The processing time of each operations performed on machine  $k$  is  $t_{ijk}$  that is greater than 0.  $C_{ijk}$  represents the completion time of the operation  $O_{ij}$ . The processing sequence of the operation is given in advance.

The following conditions need to be met when solving the FJSP:

- (1) Each machine can only process one operation at a certain time;
- (2) The carbon emissions caused by the consumption of coolant, lubricating oil, and the energy consumption of handling equipment used in the adjustment process of each job are not considered;
- (3) Once the machine starts processing, it cannot be interrupted until the operation is completed;
- (4) If there is no machine failure, the machine will not be shut down once it starts running;

(5) The processing sequence of each job process will not change during the processing;

(6) The carbon emission of the machine during processing has nothing to do with the type of workpiece, and the machine will also produce carbon emission when it is idle.

As people become more environmentally conscious, FJSP problem should incorporate not only cost and efficiency, but also low-carbon [15]. Therefore, how to obtain the optimal solution in the case of taking multiple objectives into consideration at the same time has become an urgent problem to be solved. This article mainly aims to optimize three objective functions which are respectively minimization of the maximum completion time, minimization of carbon emission and minimization of total machine load.

In formula (1),  $f_1$  represents the minimum value of the completion time ( $C_i$ ), and  $i$  represents individual. In formula (2),  $f_2$  represents the carbon emission.  $f_3$  represents minimum total machine load of machines in formula (3).

### B. FJSP MATHEMATICAL MODEL

The mathematical model of flexible job-shop scheduling problem is represented by the following related symbols:

$$f_1 = \min(\max_{1 \leq i \leq n} \sum_{i=1}^n (C_i)) \tag{1}$$

$$f_2 = \min(\sum_{k=1}^M \sum_{j=1}^{n_i} \sum_{i=1}^N (T_{ijk} X_{ijk})) \tag{2}$$

$$f_3 = \min(C_p + C_{idle}) = \alpha_e \sum_{k \in M} (\sum_{i \in N} P_{ik} + P_{idle,k} \times \sum_{i \in N} (S_{i,k} - C_{i-1,j}) + P_{i-1,ik} \sum_{i \in N} t_{i-1}) \tag{3}$$

$J = \{J_1, J_2, \dots, J_n\}$	The set of jobs ( $n$ is the number of jobs)
$M = \{M_1, M_2, \dots, M_u\}$	The set of machines ( $u$ is the number of machines)
$O_i$	The total number of operation of job $i$
$M_{ij}$	Available machines for operation $O_{i,j}$
$i, h$	Job number index, $i, h = 1, 2, \dots, n$
$j, l$	Operation number index, $j, l = 1, 2, \dots, u$
$k$	Machine number index, $k = 1, 2, \dots, u$
$O_{ik}$	It is selected to perform the operation $O_i$ on machine $k$
$P_k$	The processing power of machine $k$
$P_{idle}$	The idling power of machine $k$
$P_{ik}$	The processing power of the job $i$ in the machine $k$

$P_{i-1,ik}$	The power of machine $k$ adjusted for shifting from processing previous job to processing job $i$
$t_{i-1,ik}$	The time used by machine $k$ adjusted for shifting from processing the previous job to processing job $i$
$O_{ij}$	The $j$ th operation of the $i$ th job
$X_{ijk}$	Whether operation $O_{i,j}$ is completed on machine $k$ is 0-1 variable
$Y_{ijhkl}$	The processing sequence of operation $O_{i,j}$ and operation $O_{h,l}$ is 0-1 variable
$C_{ijk}$	The completion time of operation $O_{i,j}$ on machine $k$ ( $C_{ijk} \geq 0$ )
$T_{ijk}$	Processing time of operation $O_{i,j}$ on machine $k$ ( $T_{ijk} \geq 0$ )
$C_{time}$	Total processing time of machine
$C_p$	Carbon emission in the manufacturing process
$\alpha_e$	Carbon emission factor of electrical energy
$S_{i,k}$	Start time of processing
$C_{i,k}$	End time of processing

The relevant restraint model for FJSP problem is as follows:

$$\min C_{time} = \max C_{ik} | i \in N, k \in M \tag{4}$$

$$C_{ij} - T_{ij} \geq C_{i(j-1)} \tag{5}$$

$$\sum_{k=1}^u X_{ijk} = 1 \tag{6}$$

$$X_{ijk} = \begin{cases} 1, & \text{Operation } O_{ij} \text{ is processed on machine } k \\ 0, & \text{Other conditions} \end{cases} \tag{7}$$

$$Y_{ijhkl} = \begin{cases} 1, & \text{Operation } O_{ij} \text{ on machine } k \text{ is completed after operation } O_{hl} \\ 0, & \text{Operation } O_{ij} \text{ on machine } k \text{ is completed before operation } O_{hl} \end{cases} \tag{8}$$

$$C_{ij} = \begin{cases} \max \{C_{i(j-1)k}, S_{ijk}\} + T_{ij}, & j > 1 \\ S_{ijk} + T_{ij}, & j = 1 \end{cases} \tag{9}$$

### III. AN IMPROVED ADAPTIVE NON-DOMINATED SORTING GENETIC ALGORITHM WITH ELITE STRATEGY

#### A. ENCODING AND DECODING

With consideration of the flexibility situations of job-shop process route, this article selects the double-layer coding method, the first layer is a process-based coding scheme, which assigns the same symbol to all the operations of one

TABLE 1. IA-NSGA-ES problem coding.

Items	Operation coding									
	1	2	3	4	5	6	7	8	9	10
Job number	1	3	2	4	2	4	1	4	3	1
Sequence of machine processing	1	2	2	3	1	4	3	2	4	1
Corresponding operation	$O_{11}$	$O_{31}$	$O_{21}$	$O_{41}$	$O_{22}$	$O_{42}$	$O_{12}$	$O_{43}$	$O_{32}$	$O_{13}$
Processing time	3	2	2	3	2	4	4	2	1	3

same part, and gives definition according to the sequence of occurrence in the specified chromosome; the second layer is based on machine coding, its location value is the operation of corresponding number. The selection of machine takes not only its processing time but also machine emission into consideration.

The resulting chromosomes are:

Operation code:

$$O_{11} \rightarrow O_{31} \rightarrow O_{21} \rightarrow O_{41} \rightarrow O_{22} \rightarrow O_{42} \rightarrow O_{12} \rightarrow O_{43} \rightarrow O_{32} \rightarrow O_{13}$$

Machine code:

$$M_1 \rightarrow M_2 \rightarrow M_2 \rightarrow M_3 \rightarrow M_1 \rightarrow M_4 \rightarrow M_3 \rightarrow M_2 \rightarrow M_4 \rightarrow M_1$$

**B. POPULATION INITIALIZATION**

In the NSGA-II algorithm, the quality of the solution generated by the initial population greatly affects the solving speed and the quality sought by the algorithm, usually the scholars use the method of randomly generated initial population. On the basis of Guohui and Shijie [8], this article proposes the initial population method of considering the total completion time and carbon emission. In order to ensure population diversity, 50% of the extracted populations are generated by using a constructive heuristic method, while the other part uses random generation method.

This article uses a weight aggregation function to combine two objectives of completion time and carbon emission, in order to balance the two criteria, set a variable  $\varphi$  approaching 0 to represent higher requirement on carbon emission and a value approaching 1 to represent higher importance regarding the completion time.

**C. SELECTION OF THE OPERATION DESIGN PROGRAM**

In the genetic and evolutionary procedure of living organisms, only species with high adaptability to the living environment can have more possibilities to transmit to the next generation, the probability of heredity of less adaptable species to the next generation is relatively low. Genetic algorithm simulates this

process and sets the selection operator to select the superior and eliminate the inferior individuals in a population.

The Tournament selection method is used in this article, the application principle is: random selection of two individuals per time, individuals with higher sorting level is preferred, if the sorting level is the same, prefer the individual of larger degree of congestion, if the degree of congestion is the same, select the individual with smaller serial number.

**D. CROSSOVER OPERATION DESIGN PROGRAM**

Crossover operation refers to the exchange of partial genes between two paired chromosomes through certain method, resulting in the formation of two brand-new individuals, and plays an important role in the crossover operation of genetic algorithm. This article uses partial-mapped crossover (PMX), the steps are as follows:

Step 1: randomly select two chromosomes in a population that uses natural coding as parent generation chromosome;

Step 2: randomly generate two random numbers that meet the condition of  $0 \leq s_1 \leq s_2 \leq$  the length of the chromosome, in which the  $s_1$  is the starting point where it is selected to exchange position, and  $s_2$  is the ending point where it is selected to exchange position.

Step 3: exchange the genes selected from the two parent generations;

Step 4: perform conflict detection, regarding conflict problem of repetition in code, analogize by using mapping relationship until there is no conflict, finally resulting in a new bunch of conflict-free child generation genes.

A job to be processed is taken as an example below, the job contains nine operations, convert the operation code into machine code, randomly select any two chromosomes, that is, processing sequences [1 3 2 4 8 6 7 5] and [3 2 6 1 7 5 8 4] are taken as the parent generation, perform PMX crossover operation, assuming that  $s_1$  takes value of 2,  $s_2$  takes value of 6, the crossover procedure is shown in Figure 1.

In case of problem occurred in the procedure of conflict detection, implement mapping relation:  $2 \leftrightarrow 6 \leftrightarrow 5$ ;  $4 \leftrightarrow 1$ ;  $8 \leftrightarrow 7$ , the results obtained should be free from gene conflict phenomenon.

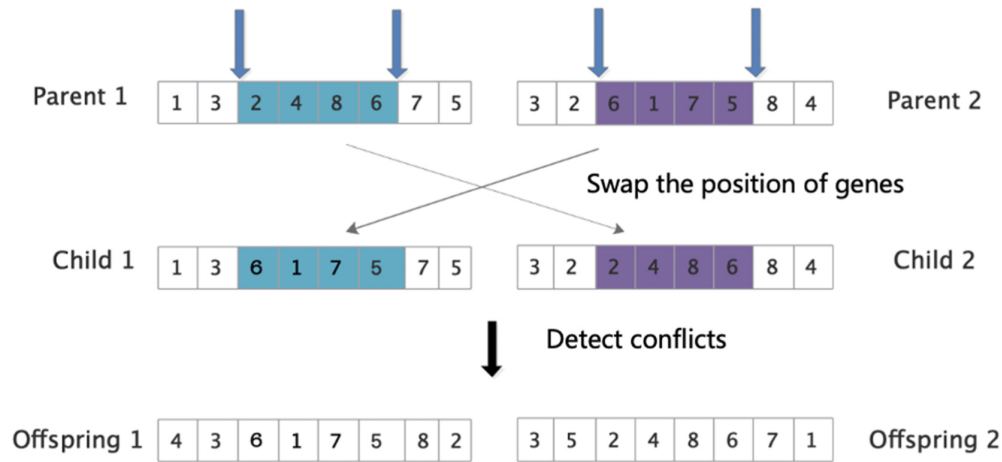


FIGURE 1. PMX crossover operation chart.

The above PMX crossover is for the cross of Operation code, the Machine code will be changed accordingly, that is, the machine with the shortest processing time is selected in its optional processing machine set.

**E. MUTATION OPERATION DESIGN PROGRAM**

Where adaptation value of sub-population generated after implementation of crossover operation no longer evolves, and the obtained results do not reach the optimal solution, it is necessary to perform a mutation operation at the moment to avoid the precocious algorithm and premature convergence. The mutation operation increases the diversity of population to some extent. The method of polynomial mutation is used in this article, and can achieve the following effect:

(1) Improves local search capability of NSGA-II. Using the crossover operator mentioned in Section D, some better individuals can already be found from a global perspective, which has approached to objective function optimal solution, but cannot be searched locally for details. The introduction of mutation operator can adjust partial gene values in individual code and get closer to optimal solution from a local perspective.

(2) Ensure population diversity. Polynomial variant operator can replace the original genes with new genes, form new individual code, improve population diversity, and prevent premature convergence of algorithm.

**F. SIMULATED ANNEALING OPTIMIZATION ELITE STRATEGY**

Simulated annealing (SA) was first proposed by Metropolis et al in 1953. It is a random optimization method based on Monte Carlo iterative solution strategy, and is based primarily on the correlation and similarity between the physical annealing procedure and combination optimization. SA starts at a higher initial temperature, uses the Metropolis sampling strategy to perform random search in the solved space field, repeats the sampling process as the temperature continuously drops, and obtains a global optimal solution of problem at the end.

Simulated annealing replacement also takes advantage of this principle, using the Metropolis principle to input child generation individuals who are superior than their parent generation into new population, otherwise leave them in the child population based on the probability of acceptance of temperature influence. This paper use the method in literature [16] to perform simulated annealing operations. The annealing rate is 0.8.

**G. IMPROVED FAST NON-DOMINATED SORTING**

The core of multi-objective optimization problem is to solve the Pareto optimal solution set. The fast non-dominated sorting [17] is to perform stratification of population according to the individual’s non-inferior solution level, so that the results gradually approach the Pareto optimal solution, this is a process in which the degree of adaptability of cycle is graded. Firstly, set the population as  $P$ , it is necessary to respectively calculate two parameters, one is the number  $n_p$  of the dominated of each individual  $P$  the other is set  $S_p$  of solution dominated by the individual, traversing the entire population, the total calculation complexity of the parameter is  $O(mN^2)$ . This article is improved on this basis, and the pseudocode is as follows:

- Step 1: calculate  $n_p$  and  $S_p$  of each individual in population;
- Step 2: put the individuals in the population that meet  $n_p = 0$  into set  $F_1$ ;
- Step 3:
  - for individual  $i \in F_1$
  - for individual  $q \in S_i$
  - $n_q = n_q - 1$ ; //eliminate domination of individuals of which Pareto rating is 1 regarding other individuals, i.e., remove individuals of which Pareto rating is 1
  - if  $n_q = 0$
  - Put the individual  $q$  in  $F_2$
  - end
  - end
  - end
- Step 4: the above steps obtain set  $F_2$  of individuals of which Pareto rating is 2, continue to repeat the step for

the individuals in  $F_2$ , and deduce the rest by analogy until complete division of population level.

### H. CROWDING DISTANCE CALCULATION

Crowding distance refers specifically to the density of individuals in a single non-dominated ranking layer after the population has been sorted as per non-dominated manner according to the dominant relationship, and the estimated chromosome congestion degree can be obtained by calculating the sum of the length and width of the rectangle made up of the two points closest to chromosome  $n$ , but excluding other individuals. The steps for calculating crowding distance are as follows:

Step 1: initial crowding distance,  $d_n = 0$ ;

Step 2: take the individuals in the single leading edge, sort according to objective function values as per ascending order, and set  $d_n = \infty$  for the individuals arranged at the first and last position;

Step 3: other individuals follow formula (11) for calculation of the crowding distance.

See formula (10), where,  $d_n$  represents the crowding distance of  $n$ th chromosome, and  $f_{n,m}$  represents the  $m$ th objective function value ( $m = 1, 2, \dots, M$ ) of  $n$  chromosome.

$$d_n = \sum_{m=1}^M (f_{n+1,m} - f_{n-1,m}) \quad (10)$$

$$chr(i) \cdot C_{(d)} = \sum_{n=1}^3 \frac{f_n(i+1) - f_n(i-1)}{f_n^{max} - f_n^{min}} \quad (11)$$

In this article, the differential method is used to set the congestion degree of two individuals as infinite, formula (11) is used to calculate the congestion degree of other individuals, in which a built-in function  $chr(i)$  is proposed in the congestion degree formula in the literature [18].

$C_{(d)}$  in the formula represents the congestion degree of chromosome  $i$ ;  $f_k$  corresponds to each objective function,  $f_1$  represents the maximum completion time,  $f_2$  represents the minimum carbon emission,  $f_3$  represents the maximum total machine load;  $f_n^{max}$  and  $f_n^{min}$  are respectively the maximum and minimum values of the  $n$ th objective of all chromosomes on one same leading edge.

### I. THE PROBABILITY OF ADAPTIVE CROSSOVER AND MUTATION

Improvements are made to the method proposed in literature [19], adaptive adjustment crossover probability  $P_c$  and mutation probability  $P_m$  are shown in formulas (12) and (13).

In which,  $f_{max}$  represents the current generation chromosome maximum adaptability value;  $f_{avg}$  represents the average adaptability value of all chromosomes of the current generation;  $f'$  represents the larger one of the two parent chromosomes involved in crossover operation;  $P_{cmax}$  represents the maximum crossover probability and takes value of 0.9 in this article,  $P_{cmin}$  represents the minimum crossover

probability and takes value of 0.6 in this article.

$$P_c = \begin{cases} P_{cmax} - \frac{P_{cmax} - P_{cmin} (f' - f_{avg})}{(f_{max} - f_{avg})}, & f' \geq f_{avg} \\ P_{cmax}, & f' < f_{avg} \end{cases} \quad (12)$$

$$P_m = \begin{cases} P_{mmax} - \frac{(P_{mmax} - P_{mmin}) (f'' - f_{avg})}{(f_{max} - f_{avg})}, & f'' \geq f_{avg} \\ P_{mmax}, & f'' < f_{avg} \end{cases} \quad (13)$$

In which,  $f_{max}$  represents the current generation chromosome maximum adaptability value;  $f_{avg}$  represents the average degree of adaptability of all chromosomes of the current generation;  $f'$  represents the larger one of adaptation value of the two parent chromosomes involved in mutation operation;  $P_{mmax}$  represents the probability of maximum mutation and takes value of 0.1 in this article,  $P_{mmin}$  represents the probability of minimum mutation and takes value of 0.001 in this article.

### J. STEPS FOR SOLVING MULTI-OBJECTIVE SHOP SCHEDULING PROBLEM ON THE BASIS OF IMPROVED ADAPTIVE NON-DOMINATED SORTING GENETIC ALGORITHM

Based on the NSGA-II algorithm, this article introduces adaptive operator by optimizing the initial population, adds the elite strategy of simulated annealing optimization, and proposes the IA-NSGA-ES algorithm to solve multi-objective flexible job-shop scheduling problem, the main steps are as follows:

Step 1: set the basic parameters in algorithm, mainly including IA-NSGA-ES parameter and workshop scheduling related parameter, such as population size, number of iterations, number optimization objectives, processing time, number of operations, carbon emission and number of jobs, etc.

Step 2: perform two types of initialization for population, randomly generate code for partial populations and generate code for other population by using construction heuristic method;

Step 3: generate operation sequence code and machine sequence code and calculate the maximum completion time, total machine load, and carbon emission;

Step 4: perform fast non-dominated sorting of the first generation of population generated, then perform the sequence of selection, crossover and mutation operation update operation, and regenerate machine sequence according to the sequence of operation;

Step 5: starting from the second generation, merge the child generation and the parent generation by using the simulated annealing elite strategy method, reserve more high-quality individuals, consider to reserve the individuals of population with small completion time and low carbon emission, and update with new parent population;

Step 6: perform fast non-dominated graded sorting for new operation population;

**TABLE 2.** Comparison with NSGA-II operation results.

Problems	IA-NSGA-ES			NSGA-II		
	$PO_1$	$t_1$	$Cp_1$	$PO_2$	$t_2$	$Cp_2$
FT06	55	10.7	15.3	57	16.6	22.6
FT10	930	36.3	336.7	930	60.2	556.8
LA01	666	6.82	106.2	666	20.6	190.9
LA03	597	9.23	235.7	599	23.9	409.8

**TABLE 3.** Comparison with adaptive NSGA-II operating results.

Problems	IA-NSGA-ES			Adaptive NSGA-II		
	$PO_1$	$t_1$	$Cp_1$	$PO_3$	$t_3$	$Cp_3$
FT06	55	10.7	15.3	55	11.5	17.4
FT10	930	36.3	336.7	930	43.5	427.6
LA01	666	6.82	106.2	—	—	—
LA03	597	9.23	235.7	597	12.7	311.0

**TABLE 4.** Comparison of test convergence speed results.

Name of functions	Name of algorithm	Mean value	Variance
ZDT1	ANSGA	0.001303	0.000418
	CDE-NAGA-II	0.0006	0.0002
	IA-NSGA-ES	0.00054	0.000203
ZDT2	ANSGA	0.001401	0.000459
	CDE-NAGA-II	0.0003	0.006
	IA-NSGA-ES	0.0082	0.00039
ZDT3	ANSGA	0.002336	0.000664
	CDE-NAGA-II	0.0033	0.0016
	IA-NSGA-ES	0.00242	0.000529

Step 7: calculate the crowding distance and congestion degree of the above-mentioned operation population;

Step 8: screen out a better Pareto solution set;

Step 9: determine whether the number of iterations is less than the maximum algebra and reaches the ending condition, terminate iteration if the condition is met; otherwise return to step 5.

#### IV. TEST SIMULATION

##### A. CALCULATION EXAMPLE SIMULATION

In order to verify that the algorithm has been improved to a certain extent, this article firstly selects different optimization

algorithms for solving multi-objective FJSP problem for comparison and verifies the calculation examples of FT06, FT10, LA01 and LA03, and algorithms include NSGA-II [14] and adaptive NSGA-II [20]. As shown in Table 2, the IA-NSGA-ES algorithm proposed in this article can effectively raise the convergence speed of non-dominated sorting genetic algorithm, and under the circumstance where the individual solutions in Pareto set are of non-dominated sorting, the optimum solution is selected, and the convergence speed is relatively raised for solving the multi-objective problem.

$PO_i$  ( $i = 1,2,3,4$ ) in the table respectively represents the optimal solution value calculated as per corresponding

TABLE 5. Flexible workshop problem for 8 jobs and 8 machines (8 × 8) at the shutter factory.

Job	Operation	Average carbon emission within processing time/unit time							
$J_1$	$O_{11}$	--	--	--	12/1.3	--	10/1.1	9/0.9	--
	$O_{12}$	17/1.2	--	--	--	17/2.9	10/2.1	--	15/1.6
	$O_{13}$	--	24/1.9	--	11/1.8	--	--	10/2	--
$J_2$	$O_{21}$	--	--	11/1.6	10/1.1	21/2.6	14/1.8	--	17/2.6
	$O_{22}$	8/0.6	12/2.2	--	--	19/1.4	11/1.2	--	--
	$O_{23}$	--	--	--	15/1.7	--	21/2.1	25/1.7	--
	$O_{24}$	--	--	--	--	18/1.7	--	9/0.9	--
	$O_{25}$	12/1.9	15/1.7	--	14/1.7	9/0.9	--	10/1.1	--
	$O_{26}$	--	9/0.8	--	7/2.1	10/2.4	8/2.1	--	--
$J_3$	$O_{31}$	--	14/2.2	--	--	--	--	17/1.6	--
	$O_{32}$	--	23/1.7	23/2	17/2.2	--	18/1.9	--	--
	$O_{33}$	--	--	20/2	22/2.1	22/2.1	--	--	22/2.1
	$O_{34}$	7/2.1	--	10/1.1	--	8/0.6	11/1.2	9/0.9	--
$J_4$	$O_{41}$	--	18/2.2	--	--	--	17/1.7	18/1.6	--
	$O_{42}$	--	--	--	10/1.3	--	--	12/1.5	--
	$O_{43}$	--	8/0.6	11/1.6	8/0.6	--	--	9/0.9	20/1.6
$J_5$	$O_{51}$	--	--	24/1.5	10/1.3	16/2.5	--	--	--
	$O_{52}$	--	--	--	18/1.8	--	--	--	19/1.4
$J_6$	$O_{61}$	--	20/2.1	--	--	--	--	22/2.1	--
	$O_{62}$	--	--	18/1.8	--	--	--	19/2.1	--
	$O_{63}$	19/2.3	--	17/1.7	16/2.4	16/2.3	--	--	18/1.8
$J_7$	$O_{71}$	--	--	16/1.9	--	17/1.6	16/2.4	--	--
	$O_{72}$	22/1.5	--	--	--	--	20/1.7	22/1.6	--
	$O_{73}$	--	--	--	--	--	18/1.8	23/1.7	--
	$O_{74}$	--	--	--	--	--	10/1.1	--	24/1.9
$J_8$	$O_{81}$	11/2.1	--	--	--	--	--	21/2.4	21/2.2
	$O_{82}$	25/1.6	11/1.7	--	--	--	--	--	--
	$O_{83}$	24/1.9	18/1.6	--	--	--	25/1.5	--	20/1.7

algorithm,  $C_{pk}$  ( $k = 1,2,3,4$ ) represents the carbon emission during the operation scheduling process according to algorithm, and “—” represents no data testing. The comparison results show that IA-NSGA-ES can obtain the optimal solution regarding the objective function problem, and the convergence speed is raised to a certain extent, and the carbon emission is reduced to a certain extent. In comparison with the carbon emission FT06 example of simulated test example, the carbon emission of this algorithm can be reduced by 12.1%-32.3%; FT10 example carbon emission can be

reduced by 21.3%-39.5%; LA01 example carbon emission is estimated to be reduced by approximately 44.4%; and LA03 example carbon emission is reduced by 24.2%-42.5% in comparison with other algorithms.

In order to verify the superiority of the algorithm, this article also makes comparison with the algorithm proposed in literature [21] (Adaptive Non-dominated sorting genetic algorithm, ANSGA) and the method in literature [22] (A Improved NSGA-II Algorithm Based on Crowding Distance Elimination Strategy CDE-NAGA-II), the comparison



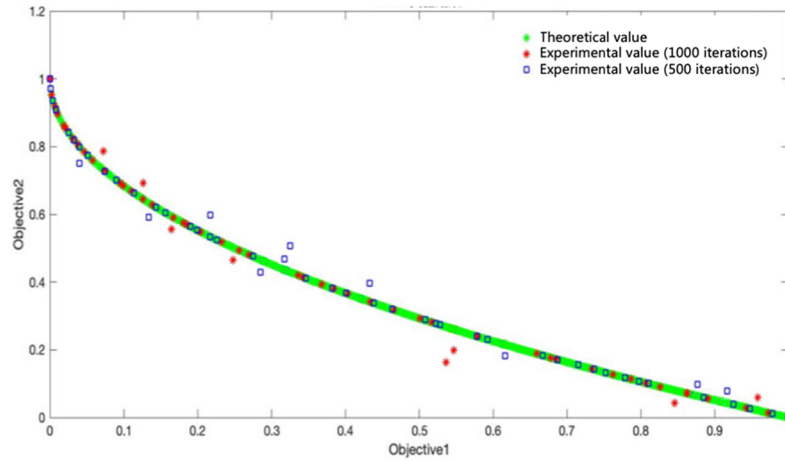


FIGURE 2. ZTD1 Pareto comparison chart.

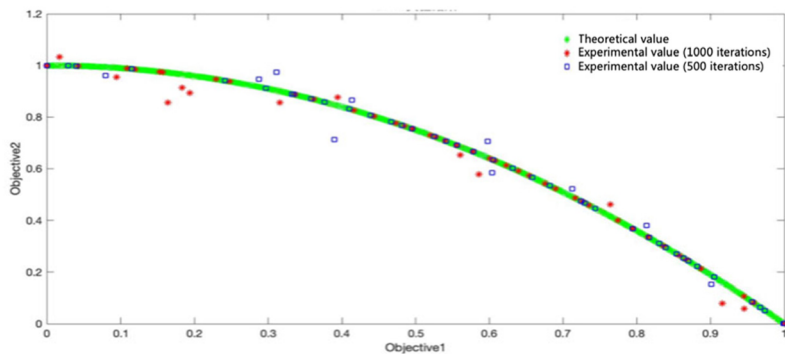


FIGURE 3. ZTD2 Pareto comparison chart.

results are shown in table 4, through the comparison of mean value and variance value, it is proved that the improved algorithm features certain enhancement in terms of convergence speed.

According to Table 4, the mean value shows that the average convergence speed has been improved to a certain extent in the test results of ZDT1 example in comparison with two comparison algorithms. The convergence time is reduced by 58.557% in comparison with ANSGA, and the convergence time is reduced by 10% in comparison with CDE-NAGA-II; there is no performance improvement in ZDT2 due to the influence of factors such as data and operating environment; and the application example of ZDT3 can guarantee that it does not lag behind others, the range differences remain around 3.596% in comparison with ANSGA, and convergence time is reduced by about 26.667% in comparison with CDE-NAGA-II. Through variance results, it is obtained through ZDT1 example that the algorithm of this article realizes reduction of 51.435% in comparison with ANSGA, and is relatively closer to CDE-NAGA-II, and the difference is 1.5%; comparison between this algorithm and ANSGA by using ZDT2 example shows that the result is reduced by 15.033%, and the result is better in comparison with CDE-NAGA-II and is reduced by 93.5%; in comparison with

ANSGA by using ZDT3 example, it is reduced by 20.331%. In comparison with CDE-NAGA-II, the deviation is reduced by 66.938%.

The ZDT1, ZDT2, and ZDT3 instances are respectively performed 500 and 1000 iterations in this paper. In an ideal state, the Pareto optimal solution obtained by the improved non-dominated sorting genetic algorithm is closer to the ideal non-dominated solution. The result comparison chart is shown in Figure 2-4.

It can be seen from the above contents that IA-NSGA-ES better applies to the circumstance where Pareto Front is convex set under the condition of 30 decision variables for solving two-objective optimization problem. In ZDT2 and ZDT3, Pareto Front is respectively non-convex set and non-connected set, and the average value is not as good as ANSGA. However, data show that in terms of dispersion degree, IA-NSGA-ES algorithm has certain advantage, and redundant data can be processed later to solve the problem that mean value rise is not high.

**B. ACTUAL WORKSHOP SIMULATION**

This article also applies IA-NSGA-ES algorithm to the actual data of a shutter factory, using the sequence of operations arranged as per optimization algorithm, the

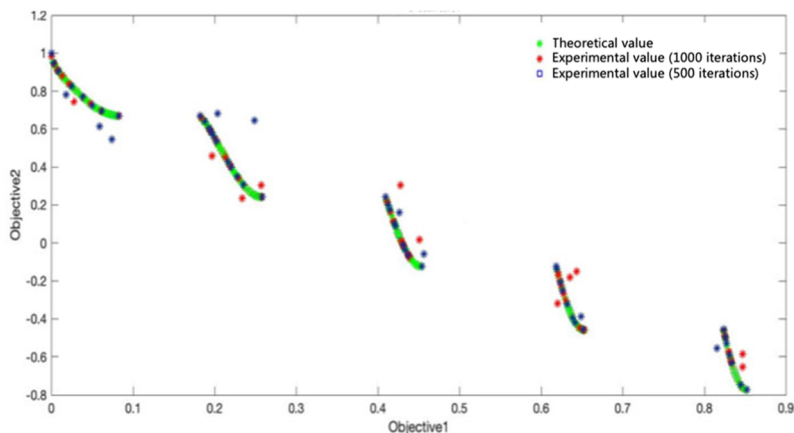


FIGURE 4. ZTD3 Pareto comparison chart.

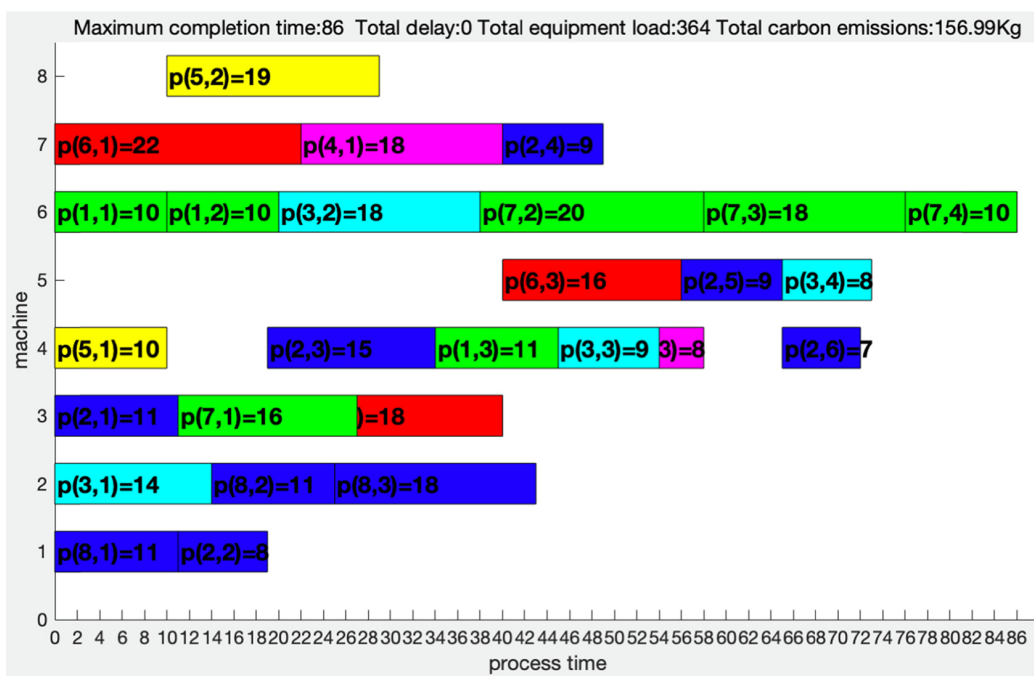


FIGURE 5. Gantt chart of the 8 x 8 problem.

processing of materials such as slat and cloth as well as the operation of shutter assembly can be completed in a shorter time.

This example is used to solve the flexible shop scheduling problem of 8 jobs and 8 machines in the shutter factory, workshop takes completion time, carbon emission and total machine load as objectives, the data of workshop is shown in Table 5, the first column represents the job number, the second column represents the operation corresponding to the job, and the numbers in the last few columns respectively represent the number of machines that can be used for processing in the operation and the average carbon emission per unit of time of processing on the machine. “-” in the table represents that the operation cannot be completed on this machine.

The example is a large-scale and complex workshop scheduling problem, the algorithm has obtained stable and high-quality optimal Pareto non-dominated solution set. Figure 5 is the optimal Gantt chart of application of the algorithm in this article in flexible workshop problem of 8 jobs and 8 machines in the shutter workshop.

After the introduction of the algorithm in this article, the processing shop has made certain improvement in aspects such as the processing time, processing efficiency and the total carbon emission of the entire processing procedure. The scheduling order obtained by the algorithm can select the machine with short processing time and relatively low carbon emission value, and in the case of emergency orders, the order specified by customer can be completed in accordance with the specified time, improving the overall work efficiency.

At the same time, the carbon emission of production process is reduced to the minimum, and green production is ensured.

## V. CONCLUSION

In this article, the existing NSGA-II algorithm is improved and mixed accordingly, not only preserving the advantages of the algorithm itself, but also complementing the disadvantages of the algorithm. Finally, through the design and development of shutter workshop scheduling system, it is verified that the improved algorithm of this article can be well combined with the system, and the corresponding scheduling program is obtained after the actual data processing, thus proving that the algorithm is effective for the actual production.

## ACKNOWLEDGMENT

The authors would like to appreciate the editor and reviewers for their helpful comments and suggestions to improve the quality of this paper.

## REFERENCES

- [1] X. B. Huang and S. L. Wei, "High-dimensional multi-objective flexible job shop scheduling considering low carbon," *J. Wut, Inf. Manage. Eng.*, vol. 41, no. 6, pp. 592–598, 2019.
- [2] P. Brucker and R. Schlie, "Job-shop scheduling with multi-purpose machines," *Computation*, vol. 45, no. 4, pp. 369–375, 1990.
- [3] Z. Cheng, Y. Tong, L.-J. Shen, and S.-S. Yu, "Double population hybrid genetic algorithm for solving flexible job-shop scheduling problem," *Comput. Eng. Des.*, vol. 37, no. 6, pp. 1636–1642, 2016.
- [4] L. Xiaotao and P. Chong, "Research of job-shop scheduling problem based on hybrid multi-intelligence genetic algorithm," *J. Beihang Univ.*, vol. 43, no. 2, pp. 410–416, 2017.
- [5] M. Chen and J. L. Li, "Genetic algorithm combined with gradient information for flexible job-shop scheduling problem with different varieties and small batches," in *Proc. MATEC Web Conf.*, vol. 95, 2017, Art. no. 10001.
- [6] D. Cinar, J. A. Oliveira, Y. I. Topcu, and P. M. Pardalos, "A priority-based genetic algorithm for a flexible job shop scheduling problem," *J. Ind. Manage. Optim.*, vol. 12, no. 4, pp. 1391–1415, 2017.
- [7] J. Yan and X. Wu, "A genetic based hyper-heuristic algorithm for the job-shop scheduling problem," in *Proc. Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, Hangzhou, China, vol. 1, 2015, pp. 161–164.
- [8] Z. Guohui and D. Shijie, "Research of low-carbon flexible job-shop scheduling problem with consideration of machine speed," *Comput. Appl. Res.*, vol. 34, no. 4, pp. 1072–1075, 2016.
- [9] R. Zhang and R. Chiong, "Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption," *J. Cleaner Prod.*, vol. 112, pp. 3361–3375, Jan. 2016.
- [10] R. L. Abreu, O. J. Cunha, A. B. Prata, and M. J. Framinan, "A genetic algorithm for scheduling open shops with sequence-dependent setup times," *Comput. Oper. Res.*, vol. 113, Jan. 2020, Art. no. 104793.
- [11] M. A. Salido, J. Escamilla, A. Giret, and F. Barber, "A genetic algorithm for energy-efficiency in job-shop scheduling," *Int. J. Adv. Manuf. Technol.*, vol. 85, nos. 5–8, pp. 1303–1314, 2016.
- [12] X. B. Pei and Y. Z. Li, "Improved genetic algorithm based on three-party game for multi-objective flexible job shop scheduling," *Int. J. Adv. Manuf. Technol.*, vol. 25, no. 4, pp. 59–69, 2020.
- [13] H. Piroozfard, K. Y. Wong, and W. P. Wong, "Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm," *Resour. Conservation Recycling*, vol. 128, pp. 267–283, Jan. 2018.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [15] W. Xiuli and S. Yangjun, "Flexible job-shop green scheduling problem with machine multi-speed," *Comput. Integr. Manuf. Syst.*, vol. 24, no. 4, pp. 862–875, 2018.
- [16] X. Shao, W. Liu, Q. Liu, and C. Zhang, "Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem," *Int. J. Adv. Manuf. Technol.*, vol. 67, nos. 9–12, pp. 2885–2901, Aug. 2013.
- [17] B. J. You, "Research on memetic algorithms for the flexible job-shop scheduling problem," Xidian Univ., Xi'an, China, 2013, pp. 23–26.
- [18] Z. Wei, "Flexible job-shop multi-objective dynamic scheduling algorithm based on rule-orientation," *Theory Pract. Syst. Eng.*, vol. 37, no. 10, pp. 2690–2699, 2017.
- [19] J. Song, C. Hou, G. Xue, and M. Ma, "Study of constellation design of pseudolites based on improved adaptive genetic algorithm," *J. Commun.*, vol. 11, no. 9, pp. 879–885, 2016.
- [20] J. Li, J. Chen, B. Xin, and L. Dou, "Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGA-II and adaptive MOEA/D: A comparison study," in *Proc. IEEE Congr. Evol. Comput.*, May 2015, pp. 3132–3139.
- [21] W. Rongbing, X. Hongyan, and G. Jun, "Adaptive non-dominated sorting genetic algorithm," *Control Decis. Making*, vol. 33, no. 12, pp. 2191–2196, 2018.
- [22] J. Liu and X. Chen, "An improved NSGA-II algorithm based on crowding distance elimination strategy," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 2, pp. 513–518, 2019.

• • •