

PRODUCTION MANAGEMENT OF HYBRID FLOW SHOP BASED ON GENETIC ALGORITHM

Chen, D.[#] & Zhao, X. R.

School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China

E-Mail: 2003500009@jsut.edu.cn ([#] Corresponding author)

Abstract

The production management of hybrid flow shop (HFS) has a great practical significance. Proper production management can improve the machine utilization and shorten the makespan in a complex production control environment. However, the relevant research has not paid enough attention to realistic constraints like multi-period control, and job transport time. To solve the problem, this paper explores the production management of HFS based on improved genetic algorithm (GA). Specifically, several assumptions were proposed for the multi-objective optimization problem of HFS production management, and new constraints like multi-period control, and job transport time were introduced to the problem. Then, the authors established a multi-objective optimization model for HFS production management, and improved the traditional GA to solve the model more rapidly and accurately. The proposed model and algorithm were proved effective through experiments.

(Received in April 2021, accepted in July 2021. This paper was with the authors 2 months for 2 revisions.)

Key Words: Genetic Algorithm (GA), Hybrid Flow Shop (HFS), Production Management

1. INTRODUCTION

The traditional production management problem of hybrid flow shop (HFS) has been proved as non-deterministic polynomial-time hard (NP-hard). The production management of HFS has a great practical significance [1, 2]. Proper production management can improve the machine utilization, reduce the waiting time of jobs, rationalize the allocation of production resources, and lower job-shop production cost [3, 4].

In actual production systems, the production methods are complicated and diversified by the changing working conditions. In response to this situation, Bukchin and Hanany [5] fully considered the capacity of batch production machines and the processing capacity of discrete machines, presented a variable batching strategy for dynamic continuous processing to minimize the makespan in HSF batch production, and solved the strategy based on the discrete water wave optimization algorithm. Viana et al. [6] designed different decoding methods to optimize machine selection and job sequence, explored the influence of processing speed on parallel machine utilization, and searched for the optimal solution under the scenario that the front and rear buffer zones of machines are limited.

If there are many types of jobs to be produced in HFS, production managers must take account of two production constraints, namely, the time setting for job sequence and the jump of operations [7-9]. Based on improved non-dominated sorting genetic algorithm (NSGA)-II, Sun et al. [10] numerically solved the multi-objective HFS production management model for the maximum production cycle and machine load, and obtained a production control plan that meets the actual needs of the enterprise. Xu and Xie [11] solved the HFS production management problem with improved differential evolution algorithm, introduced reverse learning strategy and adaptive differential factor into differential evolution, aiming to avoid the local optimum trap, and optimized the individual selection mechanism by the Metropolis criterion of simulated annealing (SA) algorithm.

The existing studies on HFS production mainly considers constraints like variable delivery period, and machine failure [12-14]. However, the relevant research has not paid

enough attention to realistic constraints like multi-period control, and job transport time. To solve the problem, this paper explores the strategy of HFS production management, and solves our multi-objective optimization model by the improved genetic algorithm (GA). Firstly, several assumptions are proposed for the multi-objective optimization problem of HFS production management, and new constraints like multi-period control, and job transport time were introduced to the problem. Next, a multi-objective optimization model was constructed for HFS production management, and the traditional GA was improved to solve the model more rapidly and accurately. Experimental results demonstrate the effectiveness of our model and algorithm.

2. MULTI-OBJECTIVE OPTIMIZATION MODEL FOR HFS PRODUCTION MANAGEMENT

Fig. 1 shows the multi-objective optimization problem for HFS production management. To simplify the problem, the following assumptions were made: (1) At the beginning of production $h = 0$, all jobs are eligible for processing, and are of the same processing priority; all machines are available for processing. (2) At any time, a machine can only process one job. (3) Each job has a fixed operation sequence; the completion time of each operation is fixed; once an operation is implemented, it cannot be interrupted. (4) The processing time of a job does not contain the time consumed to switch machines and transport the job between every two operations.

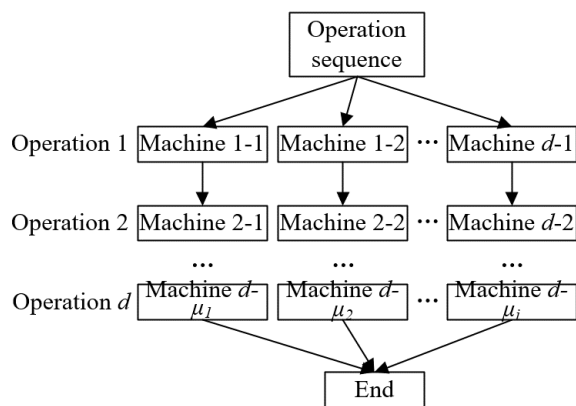


Figure 1: Multi-objective optimization problem for HFS production management.

Considering the possible problems in actual production, this paper chooses to establish the objectives of HFS production management from three perspectives: job production cycle, delivery period, and machine utilization.

During production management, a short production cycle means the jobs can be completed quickly, the production can be managed efficiently, and the enterprise can achieve a high production benefit. This paper measures the objective of production cycle with makespan g_1 . Let PR_i be the production cycle, i.e., completion time, of job I_i . Then, the objective function of production cycle, i.e., the minimum makespan, can be given by:

$$g_1 = \text{Max}PR_i \tag{1}$$

After signing the purchase contract, the job-shop needs to product the agreed number of jobs. If the jobs are not delivered within the period agreed in the contract, the enterprise should pay the buyer a default fine. Late delivery tarnishes the enterprise image, and might causes economic losses to the buyer. This paper measures the objective of delivery period with total tardiness g_2 . Let PH_i be the delivery period of job I_i . Then, the tardiness of job I_i can be expressed as:

$$g_{2-i} = \text{Max}[PR_i - PH_i, 0] \quad (2)$$

The objective of total tardiness for a production task containing M jobs can be given by:

$$g_2 = \sum_{i=1}^M \text{Max}[PR_i - PH_i, 0] \quad (3)$$

This paper measures machine utilization with the total load g_3 and maximum load g_4 of machines. Let h_{jil} be the time cost of the j^{th} operation of job I_i on the l^{th} machine; A_{jil} be the indicator of whether the j^{th} operation of job I_i is implemented on the l^{th} machine (if $A_{jil} = 1$, the operation is implemented on that machine; if $A_{jil} = 0$, the operation is not implemented on that machine). Then, machine utilization can be calculated by:

$$g_3 = \sum_{l=1}^N Q_l = \sum_{l=1}^N \sum_{i=1}^M \sum_{j=1}^{m_j} h_{jil} A_{jil} \quad (4)$$

Let Q_l be the load of the l^{th} machine. Then, the maximum machine load can be calculated by:

$$g_4 = \max_{1 \leq l \leq N} Q_l = \max \sum_{i=1}^M \sum_{j=1}^{m_j} h_{jil} A_{jil} \quad (5)$$

In this paper, the multi-objective optimization of HFS production management aims to minimize g_1 , g_2 , g_3 and g_4 simultaneously. The corresponding multi-objective optimization model can be established as:

$$OM = \text{Min}[g_1, g_2, g_3, g_4] \quad (6)$$

Let a be an N_{DV} -dimensional decision vector; b be an N_{OV} -dimensional objective vector; $g_o(a)$ be o objective functions. Normally, a multi-objective minimization problem can be expressed as:

$$OM = \text{Min}b = g_o(a) = \{g_1(a), g_2(a), \dots, g_m(a)\} \quad (7)$$

Eq. (7) shows HFS production management does not have a unique optimal solution. The production management should strike a balance between the multiple objectives, according to the actual situation. The weight vector of the multi-objective optimization of HFS production management can be given by:

$$\theta = \left\{ \theta \mid \theta \in \mathfrak{R}^{N_{ov}}, \sum_{h=1}^{N_{ov}} \theta = 1, \theta \geq 0 \right\} \quad (8)$$

Then, the multi-objective optimization problem of HFS production management can be treated as the superimposition of single-objective optimization problems:

$$\text{Min} \sum_{j=1}^{N_{ov}} \theta_j g_j(a) = g(a) = \theta_1 g_1(a) + \theta_2 g_2(a) + \dots + \theta_{N_{ov}} g_{N_{ov}}(a) \quad (9)$$

Let E_{jil} be the start time of operation GX_{jil} . Then, the time, resource, and operation constraints of HFS production can be respectively given by:

$$\begin{cases} h_{jil} \geq 0 \text{ and } h_{0il} = 0 \\ E_{jil} \geq 0 \text{ and } E_{0il} = 0 \\ E_{ji} \geq 0 \text{ and } E_{0i} = 0 \end{cases} \quad (10)$$

$$\sum_{l=1}^{N_{DV}} A_{jil} = 1, A_{jil} = 0 \text{ or } 1 \quad (11)$$

$$\sum E_{jil}A_{jil} \geq \sum \left((E_{(j-1)il} + h_{(j-1)il}) A_{(j-1)il} \right) \quad 1 \leq i \leq M, 1 \leq j \leq m_i, 1 \leq l \leq N \quad (12)$$

The production cost EC_l and profit cost EO_l of the l^{th} machine are respectively constrained by:

$$EC_l \geq 0 \text{ and } EC_0 = 0 \quad (13)$$

$$EO_l \geq 0 \text{ and } EO_0 = 0 \quad (14)$$

The number K_i of operations of job i needs to satisfy the following constraint:

$$\sum \sum A_{jil} = K_i \quad (15)$$

It is difficult to find the optimal solution to the NP-hard HFS production management problem. Therefore, this paper improves the GA based on a new encoding scheme, which ensures that the optimal solution can be found in the search space.

3. MODEL SOLVING BY IMPROVED GA

To improve the search speed and accuracy of GA, the initial population, as the start point of the search, was optimized by heuristic rules, such that the chromosomes in the population can evolve at a faster speed.

Suppose the HFS production management problem involves m jobs, n periods, and P_k ($k = 1, 2, \dots, n$) machines in period k . The chromosome length of the GA can be calculated by:

$$K = \sum_{i=1}^n (m + P_i - 1) \quad (16)$$

The initial individuals are generated gradually according to the working periods of the job-shop. For the k periods, $P_k - 1$ operation positions are generated by random, and filled with separators. The remaining m positions are filled in turn by 1 to m .

The heuristic algorithm GA gives priority to the jobs with a long production cycle, and iteratively finds the optimal local solutions to generate the job sequence. The algorithm is implemented in the following steps: (1) Sort the m jobs in descending order of production cycle; (2) Implement production management of the top 2 jobs to minimize the makespan; (3) Insert the l^{th} job ($l = 3, \dots, m$) to a possible position, and obtain the maximum processing time of the job.

Through the above steps, it is possible to obtain an operation sequence for a job on different machines in the flow line of the job-shop. In the following periods, the jobs are ranked by the first-in first-out principle, and the production task is assigned to the machine that becomes idle earlier than other machines.

To speed up the convergence to a sufficiently good solution, this paper improves the GA by replacing the hard problem of solving $n - 1$ pairs of machines with solving the simulation problem based on Johnson's rule. The production cycle of jobs can be calculated by:

$$h'_{j1} = \sum_{i=1}^n \omega_{i1} h_{ji}, h'_{j2} = \sum_{i=1}^n \omega_{i2} h_{ji} \quad (17)$$

The weight in Eq. (18) can be defined as:

$$\begin{aligned} \omega_1 &= \{ \omega_{i1} | i = 1, 2, \dots, n \} = \{ n, n-1, \dots, 2, 1 \} \\ \omega_2 &= \{ \omega_{i2} | j = 1, 2, \dots, n \} = \{ 1, 2, \dots, n-1, n \} \end{aligned} \quad (18)$$

The job sequence is adjusted with a variable weight. Further, the virtual machines are sorted by processing time based on Johnson's rule. The optimal solution is taken as the solution to the HFS production management problem. The variable weight can be defined as:

$$\begin{aligned} \omega_1 &= \{\omega_{i1} | i = 1, 2, \dots, n\} = \{n-l, n-1-l, \dots, 2-l, 1-l\} \\ \omega_2 &= \{\omega_{i2} | i = 1, 2, \dots, n\} = \{1-l, 2-l, \dots, n-1-l, n-l\} \end{aligned} \quad (19)$$

In each period of HFS production management, the production task is executed by multiple machines. Therefore, the processing time of two virtual machines can be calculated based on the mean idea load. The production cycle of jobs can be updated as:

$$h'_{j1} = \sum_{i=1}^n \frac{\omega_{i1} h_{ji}}{P_i}, h'_{j2} = \sum_{i=1}^n \frac{\omega_{i2} h_{ji}}{P_i} \quad (20)$$

Based on Johnson’s rule, the processing time of the jobs on the two virtual machines can be sorted to obtain an initial ranking of the job set. Then, the fitness function $fg_j = N^* - \max\{PR_{jn} | j = 1, 2, \dots, m\}$ of individuals in the population is defined as the difference between the total processing time N^* of all jobs and the maximum production cycle in each period. Then, N^* can be calculated by:

$$N^* = \sum_{i=1}^n \sum_{j=1}^m h_{ji} \quad (21)$$

Then, the probability of retaining a chromosome in the next generation is proportional to the fitness of that chromosome. Let $Fitness_i$ be the fitness of the i^{th} chromosome. Then, the selection probability of the chromosome can be expressed as:

$$SP(i) = \frac{Fitness(i)}{\sum_{l=1}^M Fitness(l)} \quad i = 1, 2, \dots, M \quad (22)$$

Roulette wheel selection is selected to choose chromosomes in the population. As for crossover, the partially mapped crossover (PMX) operator is combined with the single point crossover operation of inverse number vector to replace and regroup some structures of the two selected parent chromosomes, thereby producing new chromosomes. Fig. 2 explains the basic flow of the improved GA.

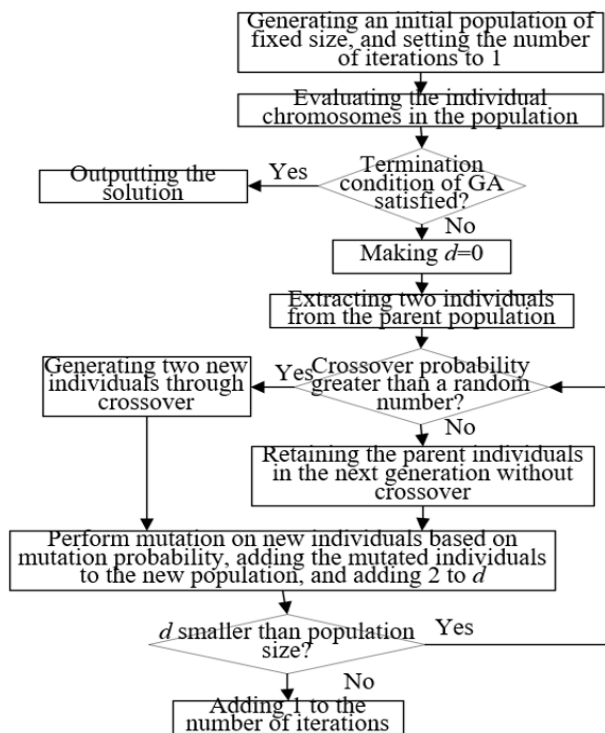


Figure 2: Basic flow of improved GA.

For a series of d elements R_1, R_2, \dots, R_d , a vector $(r_{d-1}, r_{d-2}, \dots, r_2, r_1)$ can be derived from the inverse number of every element, $0 \leq r_j \leq j, j = 1, 2, \dots, d-1$. In the other way around, a unique series of d elements R_1, R_2, \dots, R_d can be generated from a known vector $(r_{d-1}, r_{d-2}, \dots, r_2, r_1)$. The first step is to arrange a position for the d^{th} element: placing the d^{th} element in the $(r_{d-1}+1)^{\text{th}}$ blank from the right; placing the $(d-1)^{\text{th}}$ element in the $(r_{d-2}+1)^{\text{th}}$ blank from the right... placing the second element in the $(r_1+1)^{\text{th}}$ blank from the right, and placing the first element in the only remaining blank.

Suppose there are $\mu_i - 1$ separators $d + \mu_i - 1, d + \mu_i - 2, \dots, d + 1$. Then, the expanded inverse number vector of the original series can be defined as the inverse number vector obtained by inserting the separators to the whole series from 1 to d . The inserted $\mu_i - 1$ separators can be divided into μ_i groups: $R_1 R_2 / R_3 \dots R_j / \dots R_d$. The expanded inverse number vectors corresponding to these groups are $(r_{d+\mu_i-1}, r_{d+\mu_i-2}, \dots, r_d, r_{d-1}, r_{d-2}, \dots, r_2, r_1), 0 \leq r_j \leq j-1, j = 1, 2, \dots, d + \mu_i - 1$. Among them, $(r_{d+\mu_i-1}, r_{d+\mu_i-2}, \dots, r_{d+1})$ is the inverse number of $\mu_i - 1$ separators.

In the single point crossover operator of inverse number vector, there are three cases of crossover point selection: piecewise point, inverse number of elements 1 to d , and inverse number of separators. In the first two cases, the crossover can be implemented without any adjustment. In the last case, adjustment is necessary before crossover. Taking the HFS with 10 jobs and 5 machines in a period for example, the adjustment in the third case is explained below. The codes of the two parent chromosomes are expressed as:

Parent 1: [2 / 3 / 1 / 4 / 11 9 7 5 8 10 6]

Parent 2: [9 4 8 11 6 3 / 2 1 / 5 / 10 / 7]

The expanded inverse number vectors of the two chromosomes can be expressed as:

Parent 1: [11 9 8 5 6 0 7 0 1 2 0 1 2 0 0]

Parent 2: [12 6 2 4 8 6 0 10 0 10 0 6 2 0 0]

If the crossover point is at the second gene, then the new chromosomes can be encoded as:

Child 1: [12 6 2 5 6 0 7 0 1 2 0 1 2 0 0]

Child 2: [11 9 8 4 8 6 0 10 0 10 0 6 2 0 0]

Apparently, the separator inverse number of Child 1 should be adjusted. First, the inverse numbers of separators are re-sorted in descending order. Then, the adjacent inverse numbers under the condition of job number greater than machine number are adjusted to nonadjacent state.

During the PMX crossover, the crossover segment between parent and child chromosomes might be one or several periods, within one period, or across one or several periods. This paper proposes a hybrid crossover plan: implementing PMX crossover on the periods of the two ends of the crossover segment, and swapping all the periods between the two end periods.

The two parent chromosomes for crossover can be expressed as:

$CHR[x] = [4 10 / 8 6 2 6 8 / 2 / 10 0 4 2]$

$CHR[y] = [6 8 4 / 10 2 0 6 10 / 8 2 / 4 1]$

The crossover positions l_1, l_2 are randomly selected for the chromosomes, $l_1 < l_2$. If $l_1 = 0$ and $l_2 = 7$, or $l_1 = 7$ and $l_2 = 15$, new chromosomes can be obtained through direct crossover of the first segment and second segment of the two parent chromosomes. If $l_1 = 7$ and $l_2 = 15$, the first phase of the crossover can yield:

$CHR_1[x] = [4 10 / 8 6 2 6 | 8 / 2 / 10 0 4 2]$

$$CHR_1[y] = [6\ 8\ 4\ /\ 10\ 2\ 0\ |\ 6\ 10\ /\ 8\ 2\ /\ 4\ 1]$$

After the second phase of the crossover, the new chromosomes can be obtained as:

$$CHR_2[x] = [4\ 10\ /\ 8\ 6\ 2\ 6\ |\ 6\ 10\ /\ 8\ 2\ /\ 4\ 1]$$

$$CHR_2[y] = [[6\ 8\ 4\ /\ 10\ 2\ 0\ |\ 8\ /\ 2\ /\ 10\ 0\ 4\ 2]]$$

It can be seen that the new chromosomes are feasible solutions.

If $0 \leq l_1 < l_2 \leq 7$ (excluding $l_1 = 0$ and $l_2 = 7$), or if $7 \leq l_1 < l_2 \leq 15$ (excluding $l_1 = 7$ and $l_2 = 15$), PMX crossover will be implemented. Suppose $l_1 = 8$ and $l_2 = 11$, we have:

$$CHR_1[x] = [4\ 10\ /\ 8\ 6\ 2\ 6\ 8\ |\ /2\ /\ 10\ 0\ 4\ 2]$$

$$CHR_1[y] = [6\ 8\ 4\ /\ 10\ 2\ 0\ 6\ |\ 10\ /8\ |\ 2\ /4\ 1]$$

The separators are denoted as the superimposition between job number and $\mu_{i-1}, \mu_{i-2}, \dots, 2, 1$, i.e., $(d+\mu_{i-1}, d+\mu_{i-2}, \dots, d+2, d+1)$. After the second phase of the crossover, the new chromosomes can be obtained as:

$$CHR_2[x] = [4\ 10\ 7\ 8\ 6\ 2\ 6\ 8\ |\ 8\ 2\ 8\ |\ 10\ 0\ 4\ 2]$$

$$CHR_2[y] = [6\ 8\ 4\ 8\ 10\ 2\ 0\ 6\ |\ 10\ 7\ 8\ |\ 2\ 7\ 4\ 1]$$

In the third phase of crossover, the chromosomes are swapped between the two crossover positions:

$$CHR_3[x] = [4\ 10\ 7\ 8\ 6\ 2\ 6\ 8\ |\ 10\ 7\ 8\ |\ 10\ 0\ 4\ 2]$$

$$CHR_3[y] = [6\ 8\ 4\ 8\ 10\ 2\ 0\ 6\ |\ 8\ 2\ 8\ |\ 2\ 7\ 4\ 1]$$

Then, the correspondence between each data group of the two chromosomes can be determined as:

$$\begin{array}{ccc} 10 & 7 & 8 \\ \Downarrow & \Downarrow & \Downarrow \\ 8 & 2 & 8 \end{array}$$

Based on the correspondence, the child chromosomes can be formalized:

$$CHR_{S1}[x] = [4\ 10\ 7\ 8\ 6\ 2\ 6\ 8\ 10\ 7\ 8\ 10\ 0\ 4\ 2]$$

$$CHR_{S1}[y] = [6\ 8\ 4\ 8\ 10\ 2\ 0\ 6\ 8\ 2\ 8\ 2\ 7\ 4\ 1]$$

Compiling the above formulas into the original code, new chromosomes can be obtained. Replacing the data larger than d by separators, we have:

$$CHR_{S2}[x] = [4\ /7\ 8\ 6\ 2\ 6\ 8\ /7\ 8\ /0\ 4\ 2]$$

$$CHR_{S2}[y] = [6\ 8\ 4\ 8\ /2\ 0\ 6\ 8\ 2\ 8\ 2\ 7\ 4\ 1]$$

If $0 \leq l_1 \leq 7$ (excluding $l_1 = 0$) and $7 \leq l_2 \leq 15$ (excluding $l_2 = 15$), PMX crossover will be implemented. Suppose $l_1 = 3$ and $l_2 = 8$, we have:

$$CHR_1[x] = [4\ 10\ /\ |8\ 6\ 2\ 6\ 8\ |\ /2\ /10\ 0\ 4\ 2]$$

$$CHR_1[y] = [6\ 8\ 4\ |\ /10\ 2\ 0\ 6\ |\ 10\ /8\ 2\ /4\ 1]$$

The separators are denoted as the superimposition between job number and $\mu_{i-1}, \mu_{i-2}, \dots, 2, 1$, i.e., $(d+\mu_{i-1}, d+\mu_{i-2}, \dots, d+2, d+1)$. After the second phase of the crossover, the new chromosomes can be obtained as:

$$CHR_2[x] = [4\ 10\ 7\ |8\ 6\ 2\ 6\ 8\ |8\ 2\ 8\ 10\ 0\ 4\ 2]$$

$$CHR_2[y] = [6\ 8\ 4\ | \ 8\ 10\ 2\ 0\ 6\ | \ 10\ 7\ 8\ 2\ 7\ 4\ 1]$$

In the third phase of crossover, the chromosomes are swapped between the two crossover positions:

$$CHR_3[x] = [4\ 10\ 7\ | \ 8\ 10\ 2\ 0\ 6\ | \ 8\ 2\ 8\ 10\ 0\ 4\ 2]$$

$$CHR_3[y] = [6\ 8\ 4\ | \ 8\ 6\ 2\ 6\ 8\ | \ 10\ 7\ 8\ 2\ 7\ 4\ 1]$$

Then, the correspondence can be determined as:

$$\begin{array}{ccccc} 12 & 10 & 2 & 6 & 10 \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow \\ 8 & 6 & 2 & 12 & 16 \end{array}$$

Based on the correspondence, the child chromosomes can be formalized:

$$CHR_{S1}[x] = [4\ 10\ 7\ 8\ 10\ 2\ 0\ 6\ 8\ 2\ 8\ 10\ 0\ 4\ 2]$$

$$CHR_{S1}[y] = [6\ 8\ 4\ 8\ 6\ 2\ 6\ 8\ 10\ 7\ 8\ 2\ 7\ 4\ 1]$$

Replacing the data larger than d by separators, we have:

$$CHR_{S2}[x] = [4\ / \ 7\ 8\ / \ 2\ 0\ 6\ 8\ 2\ 8\ / \ 0\ 4\ 2]$$

$$CHR_{S2}[y] = [6\ 8\ 4\ 8\ 6\ 2\ 6\ 8\ / \ 7\ 8\ 2\ 7\ 4\ 1]$$

To diversify the population, mutation is a necessary operation on individual chromosomes. Firstly, chromosomes are randomly extracted by the mutation probability. Then, two integrates are randomly selected between 1 and d to characterize job number. Finally, the two job numbers are swapped to obtain a new chromosome. Suppose the current chromosome can be expressed as:

$$GT = [4\ 10\ / \ 8\ 12\ 6\ 2\ 1\ 6\ 8\ / \ 2\ / \ 10\ 4]$$

Assuming that job 2 and job 8 are swapped:

$$GT = [4\ 10\ / \ 8\ 12\ 6\ (2)\ 1\ 6\ (8)\ / \ 2\ / \ 10\ 4]$$

A new chromosome can be obtained by exchanging the positions of the two jobs:

$$GT = [4\ 10\ / \ 8\ 12\ 6\ 8\ 1\ 6\ 2\ / \ 2\ / \ 10\ 4]$$

4. EXPERIMENTAL VERIFICATION

Our algorithm was realized on Visual C++. Since the multi-objective optimization of HFS production management is an NP-hard scheduling problem, this paper designs contrastive experiments based on actual production management problems.

Table I: Experimental data on the production management of 12 jobs in 6 periods.

Random seeds	$CP = 0.8, MP = 0.01$	$CP = 0.6, MP = 0.01$	$CP = 0.6, MP = 0.005$
0	4724	4923	4873
1	5139	4582	5192
2	5162	4735	4751
3	5125	4841	5329
4	5139	5329	4713
5	4781	4893	4795
6	4453	4896	4869
7	4871	4792	4534
8	4665	5461	5217
9	4982	4732	4782

The parameters of the improved GA were configured as follows: the population size, 100; the maximum number of iterations, 1,000; crossover probability CP , 0.6 and 0.8; mutation probability MP , 0.001, 0.005, and 0.01. Under the random seeds of 0-9, each production data was tested 10 times. Table I compares the experimental data on the production management of 12 jobs in 6 periods, each of which has 4 machines.

Table II compares the experimental data on the production management of 14 jobs in 4 periods, each of which has 3 machines.

Table II: Experimental data on the production management of 14 jobs in 4 periods.

Random seeds	$CP=0.8, MP=0.01$	$CP=0.6, MP=0.01$	$CP=0.6, MP=0.005$
0	4127	4329	4365
1	4351	4730	4572
2	4738	4813	4821
3	4532	4528	4519
4	4574	4235	4632
5	4539	4319	4524
6	4620	4521	5241
7	4786	4637	4386
8	4921	4264	4218
9	4654	4209	4679

Table III compares the experimental data on the production management of 13 jobs in 5 periods, each of which has 3 machines.

Table III: Experimental data on the production management of 13 jobs in 5 periods.

Random seeds	$CP=0.8, MP=0.01$	$CP=0.6, MP=0.01$	$CP=0.6, MP=0.005$
0	4750	5193	5316
1	5026	5427	4882
2	5472	5614	5126
3	5640	4826	5343
4	5234	5208	5628
5	5729	5239	5725
6	5318	4712	4839
7	5272	5126	5512
8	5323	5823	4831
9	5624	5464	5680

Table IV compares the experimental data on the production management of 19 jobs in 6 periods, each of which has 4 machines.

Table IV: Experimental data on the production management of 19 jobs in 6 periods.

Random seeds	$CP=0.8, MP=0.01$	$CP=0.6, MP=0.01$	$CP=0.6, MP=0.005$
0	892	1045	932
1	876	928	916
2	881	1036	923
3	1032	962	837
4	1024	945	934
5	953	1023	982
6	912	952	923
7	956	924	874
8	970	950	956
9	895	963	973

Fig. 3 compares the trend of the objective function value of traditional GA and improved GA in solving the production management problems of Tables I and IV. With the growing number of iterations, the improved GA achieved better solutions than the traditional GA in handling the problems with 12 and 19 jobs.

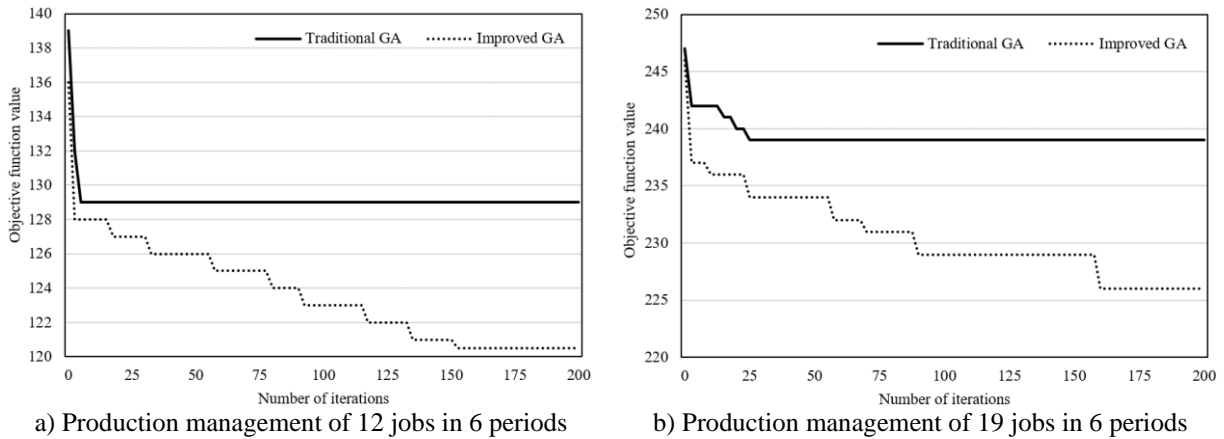


Figure 3: Trends of objective function value in solving problems of different scales.

Table V lists ten sets of optimal solutions obtained by the improved GA, with production cycle, total tardiness, and machine utilization as objectives. Fig. 4 offers the Gantt charts of HFS production management before and after multi-objective optimization.

Table V: Sets of optimal solutions.

Serial number	Production cycle	Total tardiness	Machine utilization
0	18	6	58
1	17	8	67
2	17	10	54
3	18	12	65
4	19	6	43
5	18	10	64
6	17	8	42
7	18	6	62
8	17	8	63
9	17	9	64

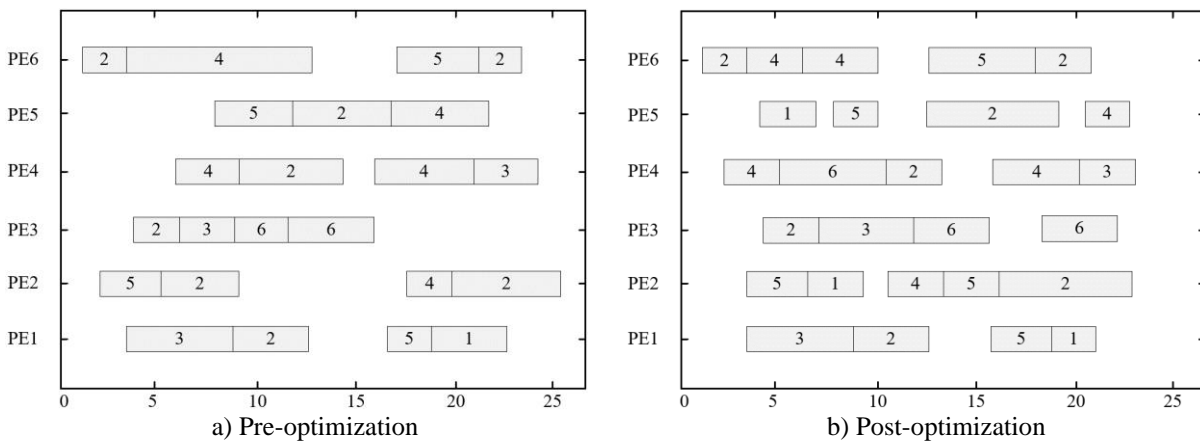


Figure 4: Gantt charts before and after multi-objective optimization.

As shown in Fig. 4, the job sequence changed through the multi-objective optimization, that is, the job-shop had different processing routes before and after the optimization. The

optimal solutions obtained by our algorithm changed with the routes. Therefore, the proposed algorithm can support flexible production management of actual HFSs. To verify the effectiveness of our algorithm in actual production, some source data of the production management of 10 machines and 50 operations are given in Table VI.

Table VI: Source data of production management experiment.

Operations	PR_{11}	PR_{21}	PR_{31}	PR_{41}	PR_{12}	PR_{22}	PR_{32}	PR_{42}	PR_{13}	PR_{23}	PR_{33}	PR_{34}	PR_{14}
PE_1	3	5	3	11	5	7	9	10	4	7	7	2	7
PE_2	4	2	7	6	7	12	4	3	8	15	3	4	3
PE_3	7	5	3	6	4	3	9	7	7	12	2	11	6
PE_4	10	5	6	12	2	9	5	5	2	20	9	2	5
PE_5	4	9	7	10	8	6	8	9	11	10	7	6	3
PE_6	6	11	10	5	9	4	9	2	2	3	4	9	5
PE_7	3	5	16	2	6	7	13	2	2	3	2	10	4
PE_8	9	12	11	9	11	15	12	10	3	18	3	4	7
PE_9	10	5	4	6	10	11	9	6	3	2	10	5	6
PE_{10}	5	4	3	5	2	3	13	4	5	7	5	7	4

5. CONCLUSIONS

This paper tries to develop a suitable strategy for HFS production management. After making assumptions for the multi-objective optimization of HFS production management, the authors introduced new constraints like multi-period control, and job transport time to establish a multi-objective optimization model for HFS production management. Contrastive experiments were designed to simulate production management problems with different jobs, periods, and machines, and the relevant data were summarized. To improve the search speed and accuracy of the solving algorithm, the traditional GA was improved for the proposed model. Based on actual production management problems, the authors compared the trends of the objective function value of traditional GA and improved GA, and confirmed the superiority of the improved GA in solution quality. Finally, the effectiveness of our algorithm in actual production was further verified by the Gantt charts of HFS production management before and after multi-objective optimization, and some source data of the production management of 10 machines and 50 operations.

ACKNOWLEDGEMENT

This paper was supported by Jiangsu Province Industry-University-Research Cooperation Project (No. BY2018191).

REFERENCES

- [1] Istokovic, D.; Perinic, M.; Vlatkovic, M.; Brezocnik, M. (2020). Minimizing total production cost in a hybrid flow shop: a simulation-optimization approach, *International Journal of Simulation Modelling*, Vol. 19, No. 4, 559-570, doi:[10.2507/IJSIMM19-4-525](https://doi.org/10.2507/IJSIMM19-4-525)
- [2] Awaga, A. L.; Xu, W.; Liu, L.; Zhang, Y. (2020). Evolutionary game of green manufacturing mode of enterprises under the influence of government reward and punishment, *Advances in Production Engineering & Management*, Vol. 15, No. 4, 416-430, doi:[10.14743/apem2020.4.375](https://doi.org/10.14743/apem2020.4.375)
- [3] Yahouni, Z.; Mebarki, N.; Sari, Z. (2019). Evaluation of a new decision-aid parameter for job shop scheduling under uncertainties, *RAIRO – Operations Research*, Vol. 53, No. 2, 593-608, doi:[10.1051/ro/2017073](https://doi.org/10.1051/ro/2017073)
- [4] Zarrouk, R.; Bennour, I. E.; Jemai, A. (2019). A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem, *Swarm Intelligence*, Vol. 13, No. 2, 145-168, doi:[10.1007/s11721-019-00167-w](https://doi.org/10.1007/s11721-019-00167-w)

- [5] Bukchin, Y.; Hanany, E. (2020). Decentralization cost in two-machine job-shop scheduling with minimum flow-time objective, *IISE Transactions*, Vol. 52, No. 12, 1386-1402, doi:[10.1080/24725854.2020.1730528](https://doi.org/10.1080/24725854.2020.1730528)
- [6] Viana, M. S.; Morandin Junior, O.; Contreras, R. C. (2020). A modified genetic algorithm with local search strategies and multi-crossover operator for job shop scheduling problem, *Sensors*, Vol. 20, No. 18, Paper 5440, 32 pages, doi:[10.3390/s20185440](https://doi.org/10.3390/s20185440)
- [7] Wang, C.; Li, Y.; Li, X. (2021). Solving flexible job shop scheduling problem by a multi-swarm collaborative genetic algorithm, *Journal of Systems Engineering and Electronics*, Vol. 32, No. 2, 261-271, doi:[10.23919/JSEE.2021.000023](https://doi.org/10.23919/JSEE.2021.000023)
- [8] Vela, A.; Cruz-Duarte, J. M.; Ortiz-Bayliss, J. C.; Amaya, I. (2021). Tailoring job shop scheduling problem instances through unified particle swarm optimization, *IEEE Access*, Vol. 9, 66891-66914, doi:[10.1109/ACCESS.2021.3076426](https://doi.org/10.1109/ACCESS.2021.3076426)
- [9] Ahmadian, M. M.; Salehipour, A.; Cheng, T. C. E. (2021). A meta-heuristic to solve the just-in-time job-shop scheduling problem, *European Journal of Operational Research*, Vol. 288, No. 1, 14-29, doi:[10.1016/j.ejor.2020.04.017](https://doi.org/10.1016/j.ejor.2020.04.017)
- [10] Sun, J.; Zhang, G.; Lu, J.; Zhang, W. (2021). A hybrid many-objective evolutionary algorithm for flexible job-shop scheduling problem with transportation and setup times, *Computers & Operations Research*, Vol. 132, Paper 105263, 15 pages, doi:[10.1016/j.cor.2021.105263](https://doi.org/10.1016/j.cor.2021.105263)
- [11] Xu, L.-Z.; Xie, Q.-S. (2021). Dynamic production scheduling of digital twin job-shop based on edge computing, *Journal of Information Science & Engineering*, Vol. 37, No. 1, 93-105, doi:[10.6688/JISE.202101_37\(1\).0007](https://doi.org/10.6688/JISE.202101_37(1).0007)
- [12] Sarsur, D. C.; Pena, P. N.; Takahashi, R. H. C. (2020). Automatic translation of blocking flexible job shop scheduling problems to automata using the supervisory control theory, *IFAC Papers OnLine*, Vol. 53, No. 4, 89-94, doi:[10.1016/j.ifacol.2021.04.009](https://doi.org/10.1016/j.ifacol.2021.04.009)
- [13] Rodrigues, R. P.; de Pinho, A. F.; Sena, D. C. (2020). Application of hybrid simulation in production scheduling in job shop systems, *Simulation*, Vol. 96, No. 3, 253-268, doi:[10.1177/0037549719861724](https://doi.org/10.1177/0037549719861724)
- [14] Jiang, E.-D.; Wang, L.; Peng, Z.-P. (2020). Solving energy-efficient distributed job shop scheduling via multi-objective evolutionary algorithm with decomposition, *Swarm and Evolutionary Computation*, Vol. 58, Paper 100745, 16 pages, doi:[10.1016/j.swevo.2020.100745](https://doi.org/10.1016/j.swevo.2020.100745)