Flexible flow shop scheduling problem with setup times and blocking constraint via genetic algorithm and simulation

Meng Zhang

Liaoning Equipment Manufacture College of Vocational Technology, Liaoning Shenyang ,China.
564202513@qq.com

Abstract. Flexible flow shop scheduling problem (FFSP) is recognized as an important class of problems in manufacturing systems. To consider more actual production factors, such as transportation time, setup times, blocking constraint, etc., we solved the FFSP using the improved genetic algorithm (GA) and discrete event simulation (DES). Firstly, a mathematical model for FFSP with the objective of minimizing total completion time was established. Besides, a GA mixed with Palmer heuristic algorithm was proposed to solve the mathematical model. Moreover, a DES software-plant simulation was used to establish a more realistic production model, and an actual production workshop was modeled and optimized. After optimization, the total completion time and equipment utilization were greatly improved compared with the original scheduling. It shows the effectiveness and rationality of the proposed method to solve the actual FFSP.

Keywords: flexible flow shop scheduling problem (FFSP); genetic algorithm (GA); Plant Simulation; blocking; setup times.

1. Introduction

The flexible flow shop scheduling problem (FFSP) is the promotion of the flow shop scheduling problem (FSP). The FFSP is considered to have a few parallel production machines or lines. Flexible flow shop methodologies have been used in different industries such as semiconductor, automotive, chemical, etc. [1]

The FFSP has attracted considerable attention since it was identified in the 1970s [2] and many algorithms are proposed to solve the FFSP. Zhang et al. [3] proposed an ant colony optimization for permutation flow shop scheduling problem. Han et al. [4] studied the NSGA-II algorithm for multi-objective scheduling problem in a hybrid flow shop. And iterated greedy algorithms (IG) was used for the HFSP with total flow time criterion [5]. Moreover, the variable neighborhood search algorithm and meta-heuristics [6] was also used to solve the FFSP.

In addition to these algorithms, the genetic algorithms (GA) have been widely used in solving the FFSP for its powerful global search capabilities. Mousavi et al. [7] solved a bi-objective hybrid flow shop scheduling problem with the objective of minimizing the maximum completion time (makespan) and total tardiness using GA. Dabiri et al. [8] studied the multi-machine flow shop scheduling problems with rejection using GA. Wu et al. [9] proposed a hybrid non-dominated sorting genetic algorithm with variable local search to solve a multi-objective flexible flow shop scheduling problem which considers variable processing time. Luo et al. [10] proposed a dual heterogeneous island genetic algorithm for solving large size flexible flow shop scheduling problems. Azami [11] proposed a GA with a new
cross-over operator and a heuristic algorithm to solve the FFSP, and considered the buffer capacity and the limited waiting time between the stations.

Discrete event simulation (DES) is a discrete-state and event-driven method for systems whose behavior is characterized by abrupt changes in the value of its state [12]. DES can efficiently build the simulation models of production systems considering more practical production factors. The Plant simulation software package is one of the DES software. Plant simulation has powerful logistics modeling and simulation capabilities to fully simulate the actual production process and can achieve precise and flexible control for different production conditions[13].

Based on the above literature, it can be found that various algorithms have been proposed to solve the FFSP. However, when using these intelligent algorithms, many actual production factors, such as blocking constraint and setup times are often neglected. Although a few researchers have considered some realistic factors[14], the establishment and solution of mathematical models are complicated. Moreover, although DES can efficiently establish a simulation model for production processes, few researchers have used the intelligent algorithms and DES in combination to solve the FFSP.

In this paper, the accurate simulation model of FFSP is established by Plant Simulation software and optimized by the improved GA. The actual production factors, such as the transportation time, the blocking constraint, and the setup times were considered in the simulation model. In addition, Palmer heuristic algorithm was used to generate an initial population, thereby speeding up the GA convergence. Finally, an actual flexible flow shop was used as a case to demonstrate the effectiveness of the proposed method in solving the FFPS.

2. Problem description and mathematical model

FFSP can be described as follows: n parts need to be processed, each part have m steps, and at least one of these processes contains parallel stations. The purpose of the study was to determine the order in which each part was machined at each station to minimize the total completion time of the batch.

The objective function is formulated as:

\[ \min Z = C_{\text{max}} \]  

Subject to:

\[ \sum_{k \in K_j} x_{i,j,k} = 1, \quad \forall i, j \]  

\[ \sum_{i=1}^{n} y_{i,k,t} = x_{i,j,k}, \quad \forall i, j, k \in K_j \]  

\[ \sum_{i=1}^{n} y_{i,k,t} \leq 1, \quad \forall k, t \]  

\[ \sum_{i=1}^{n} y_{i,k,t} \geq \sum_{i=1}^{n} y_{i,k,t+1}, \quad \forall k, t < n \]  

\[ F_{k,j} = B_{i,j} + \sum_{k \in K_j} (P_{i,k} \times x_{i,j,k}), \quad \forall i, j \]  

\[ F_{k,t} = S_{k,t} + \sum_{i=1}^{n} \sum_{j=1}^{m} (P_{i,k} \times y_{i,j,k}), \quad \forall k, t \]  

\[ S_{k,t} \leq B_{i,j} + M \times (1 - y_{i,k,t}), \quad \forall i, j, k \in K_j, t \]  

\[ S_{k,t} + M \times (1 - y_{i,k,t}) \geq B_{i,j}, \quad \forall i, j, k \in K_j, t \]  

\[ F_{k,t} \leq S_{k,t}, \quad \forall k, t < n \]  

\[ E_{i,j} \leq B_{i,j}, \quad \forall i, j < S \]  

\[ C_{\text{max}} \geq E_{i,S}, \quad \forall i \]
\[ x_{i,j,k} = \begin{cases} 1, & \text{if } j\text{th job of part } i \text{ is processed in machine } k \\ 0, & \text{otherwise} \end{cases} \] (13)

\[ y_{i,k,t} = \begin{cases} 1, & \text{if part } i \text{ is processed in the } t\text{th job of machine } k \\ 0, & \text{otherwise} \end{cases} \] (14)

Where \( i \) denotes part, \( I \) denotes a set of parts, \( J \) denotes a set of jobs, \( K \) denotes a set of machines, \( P_{ik} \) denotes the processing time of part \( I \) in the \( j \)th event, \( S_{kt} \) denotes the start time of \( t \)th event in machine \( k \); \( F_{kt} \) denotes the end time of \( t \)th event in machine \( k \); \( C_{\text{max}} \) denotes the maximum completion time; \( B_{i,j} \) denotes the start time of the \( j \)th event of part \( i \); \( E_{i,j} \) denotes the end time of the \( j \)th event of part \( i \); \( V \) is a sufficiently large positive number.

Eq. 2 ensures that any process of the part can only be processed once; Eq. 3 ensures that whether a part is processed on this machine is determined by the assignment scheme; Eq. 4 ensures that a machine can only process one part at a time; Eq. 5 ensures that each machine is processed in the specified order; Eq. 6 and Eq. 7 ensure that the end time of each operation equals the start time plus the processing time; Eq. 8 and Eq. 9 ensure that the start time of the machine is equal to the start time of the part processed in the machine; Eq. 10 ensures that the end time of the previous event in any machine is not greater than the start time of subsequent events; Eq. 11 ensures that the end time of the previous event of a part is not greater than the start time of the subsequent event; Eq. 12 represents the Makespan.

3. Design of Genetic Algorithms

GA is a highly parallel, stochastic and adaptive optimization algorithm. It represents a solution and an objective function of the problem using chromosomes and the fitness of these chromosomes. The basic elements of GA include chromosome coding, population initialization, decoding, selection, crossover and mutation. Firstly, a group of initial population representing the solution of the problem is generated by coding, and the chromosomes with good adaptability are selected according to its fitness value. These chromosomes are regarded as the paternal generation. Then, through the crossover and mutation, excellent genes are inherited to form offspring population. Through the continuous evolution of the chromosome, the population eventually converges to the optimal individual and gets the optimal solution of the problem.

3.1. Coding and Decoding

The coding method was based on process expression, the assembly sequence of each product is encoded as a chromosome. Each gene represents a process, and for the same part, the sequence in the chromosome represents the assembly sequence of the parts. After decoding, \( O_{ijk} \) indicates that the \( j \)th process of product \( i \) operates on the workstation \( k \). This coding and decoding method is illustrated with a scheduling problem of three production workstations with three products. The operation schedule of the product is shown in Table 1. Each product is processed on three workstations in turn: SP1, SP2 and SP3. If a scheduling code is \([1 \ 2 \ 1 \ 1 \ 3 \ 2 \ 3 \ 2 \ 3]\), it can be decoded as \([O_{111}, O_{211}, O_{122}, O_{133}, O_{311}, O_{222}, O_{322}, O_{233}, O_{333}]\), according to the encoding and decoding rules. The decoding process and scheduling schematics are shown in Figure 1 and Figure 2, respectively.

| Product | Processing Time |
|---------|-----------------|
|         | SP1 | SP2 | SP3 |
| J1      | 2   | 2   | 4   |

Table 1. Processing time of three production workstations with three products.
3.2. Initial population.

The generation of an initial population is usually random. In this paper, a Palmer heuristic algorithm was used to produce an initial population. The Palmer algorithm ranks the jobs in the order that the slope index does not increase. The slopes of the parts are as follows:

\[ \lambda_i = \sum_{j=1}^{M} \left[ \frac{j-(m+1)}{2} \right] t_{ij} \quad (i = 1, 2, ..., n) \]

Where \( M \) denotes the number of machines; \( t_{ij} \) denotes the processing time of part \( J_i \) on machine \( M_j \).

3.3. Selection.

Selection operations select a number of better individuals from the current population with a certain probability. Individuals with high fitness are more likely to be selected. Selection operation makes the high-performance individuals have a greater survival probability, avoids the deletion of effective genes, and improves the global convergence and computational efficiency of the solution.

3.4. Crossover.

Crossover operation selects two individuals from the parent population for gene exchange to form a new generation of individuals. Partial mapping cross (PMX) approach was used in this paper. Firstly, two random points are randomly inserted into the two parents to exchange the fragments between the intersections of the parents. Then, for the new individuals formed after the exchange, inspect whether the gene exchanged conflicts with the genes outside the chromosome crossing point. And the conflicting genes will be modified by partial mapping.

4. Simulation Modeling and Optimization

An actual production workshop was taken as an example, and the plant simulation software was used to establish the simulation model of the workshop production. The scheduling optimization model of the
assembly workshop was established, according to the assembly process and plant layout. As shown in Figure 3, the assembly shop includes a material shop, two assembly shop, and a packaging workshop. The AS workstation has two parallel workstation s, and the T workstation has three parallel workstations. The assembly assignment on the parallel machine is assigned according to the machine's no idle principle. The Automated Guided Vehicle (AGV) is used for transportation between the workshops, and the transmission lines are used between workstations. Five types of products are produced, and the operation time for each product are shown in Table 2.

![Figure 3. Production simulation model of an assembly shop.](image)

### Table 2. The operation time (s) for each product.

| Product | SP1 | SP2 | SP3 | SP4 | AS1 | AS2 | T1 | T2 | T3 | T4 | P1 | P2 |
|---------|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|
| J1      | 63  | 75  | 81  | 76  | 297 | 246 | 400| 328| 247| 122| 215| 315|
| J2      | 88  | 90  | 74  | 122 | 143 | 372 | 303| 347| 267| 143| 285| 250|
| J3      | 152 | 130 | 105 | 116 | 190 | 262 | 352| 273| 413| 230| 175| 470|
| J4      | 110 | 133 | 125 | 132 | 375 | 327 | 250| 373| 237| 185| 195| 310|
| J5      | 133 | 115 | 113 | 141 | 287 | 205 | 377| 283| 340| 113| 263| 390|

There are setup times when the AS workstation processes different parts. The required setup times are shown in Table 3 and Table 4.

### Table 3. Setup times (s) for different products in AS1 workstation.

| Product | J1 | J2 | J3 | J4 | J5 |
|---------|----|----|----|----|----|
| idle    | 18 | 13 | 19 | 17 | 14 |
| J1      | 0  | 11 | 21 | 5  | 6  |
| J2      | 12 | 0  | 17 | 5  | 5  |
| J3      | 5  | 5  | 0  | 10 | 9  |
| J4      | 4  | 5  | 4  | 0  | 4  |
| J5      | 5  | 8  | 5  | 5  | 0  |
### Table 4. Setup times (s) for different products in AS2 workstation.

| Product | J1 | J2 | J3 | J4 | J5 |
|---------|----|----|----|----|----|
| idle    | 13 | 16 | 15 | 22 | 19 |
| J1      | 0  | 15 | 16 | 5  | 9  |
| J2      | 6  | 0  | 5  | 4  | 7  |
| J3      | 7  | 4  | 0  | 9  | 6  |
| J4      | 8  | 7  | 4  | 0  | 8  |
| J5      | 8  | 8  | 5  | 6  | 0  |

5. **Establishment of the simulation model**

An actual production workshop was taken as an example, and the plant simulation software was used to establish the simulation model of the works. Assuming that the quantity of each product is one, the simulation is run according to the initial production sequence \((J_1, J_3)\). After the simulation, the total assembly completion time is 40 minutes and 4 seconds, and the Gantt chart of each workstation is shown in Figure 4.

![Gantt chart under the initial production sequence](image)

The scheduling simulation model was optimized by the improved GA. The number of generations is 20, the generation size is 30, and the mutation probability is 0.1. After optimization, the total completion time is 36 minutes and 15 seconds, and the optimal assembly scheduling sequence is \((J_2, J_1, J_4, J_3)\). The evolution curve and the fitness of the offspring are shown in Figure 5 and Figure 6, respectively. It can be seen from the figure that as the evolutionary algebra increases, the result tends to be optimal and converges to the optimal scheduling sequence after the seventh generation. The Gantt chart under optimal production sequence is shown in Figure 7.

![Performance Graph](image)
Figure 5. Evolution curve of the GA.

Figure 6. Fitness of the offspring.

Figure 7. Gantt chart under the optimal production sequence.

In addition, under stable production conditions, the facility utilization corresponding to the initial production sequence and the optimized production sequence are shown in Figure 8 and Figure 9, respectively. It can be seen from the figure that after optimization the utilization of each workstation is increased.

Figure 8. Facility utilization under the initial production sequence.
6. Conclusions
To obtain a more realistic simulation model and optimization results for FFSP, this paper uses a combination of improved genetic algorithm and discrete event simulation. The Palmer heuristic algorithm was used to obtain the initial solution, the GA was used to obtain the optimal solution, and the Plant Simulation software was used for production simulation. The process-based coding method was adopted to prevent the generation of illegal chromosomes, and encoding and decoding are simpler and more intuitive. In addition, more realistic production factors, such as transportation time, setup times and blocking constraints are considered in the Plant Simulation software. After optimization, total completion time is reduced and equipment utilization is increased. It shows that the optimization method proposed in this paper can effectively solve the FFSP with more realistic constraints, and can provide decision guidance for the scheduling optimization in actual production conditions.

References
[1] LEE T-S, LOONG Y-T. A review of scheduling problem and resolution methods in flexible flow shop [J]. International Journal of Industrial Engineering Computations, 2019, 67-88.
[2] CHOI S H, WANG K. Flexible flow shop scheduling with stochastic processing times: A decomposition-based approach [J]. Computers & Industrial Engineering, 2012, 63(2): 362-73.
[3] ZHANG Y, YU Y, ZHANG S, et al. Ant colony optimization for Cuckoo Search algorithm for permutation flow shop scheduling problem [J]. Systems Science & Control Engineering, 2018, 7(1): 20-7.
[4] HAN Z, WANG S, DONG X, et al. Improved NSGA-II algorithm for multi-objective scheduling problem in hybrid flow shop [M]. Innovative Techniques and Applications of Modelling, Identification and Control. Springer. 2018: 273-89.
[5] ÖZTOP H, TASGETIREN M F, ELIYI D T, et al. Iterated greedy algorithms for the hybrid flowshop scheduling with total flow time minimization [J]. 2018, 379-85.
[6] ABDOLLAHPOUR S, REZAIAN J. Two new meta-heuristics for no-wait flexible flow shop scheduling problem with capacitated machines, mixed make-to-order and make-to-stock policy [J]. Soft Computing, 2016, 21(12): 3147-65.
[7] MOUSAVI S M, MAHDAVI I, REZAEIAN J, et al. An efficient bi-objective algorithm to solve re-entrant hybrid flow shop scheduling with learning effect and setup times [J]. Operational Research, 2016, 18(1): 123-58.
[8] DABIRI M, DARESTANI S A, NADERI B. Multi-machine flow shop scheduling problems with rejection using genetic algorithm [J]. International Journal of Services and Operations Management, 2019, 32(2): 158-72.

[9] WU X, SHEN X, CUI Q. Multi-Objective Flexible Flow Shop Scheduling Problem Considering Variable Processing Time due to Renewable Energy [J]. Sustainability, 2018, 10(3): 841.

[10] LUO J, EL BAZ D. A Dual Heterogeneous Island Genetic Algorithm for Solving Large Size Flexible Flow Shop Scheduling Problems on Hybrid Multicore CPU and GPU Platforms [J]. Mathematical Problems in Engineering, 2019, 1-13.

[11] AZAMI A, DEMIRLI K, BHUIYAN N. Scheduling in aerospace composite manufacturing systems: a two-stage hybrid flow shop problem [J]. The International Journal of Advanced Manufacturing Technology, 2018, 95(9-12): 3259-74.

[12] SILVA M. On the history of Discrete Event Systems [J]. Annual Reviews in Control, 2018, 213-22.

[13] FERRO R, CORDEIRO G A, ORDOñEZ R E C. Dynamic Modeling of Discrete Event Simulation [J]. 2018, 248-52.

[14] JANARDHANAN M N, LI Z, BOCEWICZ G, et al. Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times [J]. Applied Mathematical Modelling, 2019, 256-70.