Solving flow shop scheduling problem based on improved non-dominated genetic algorithm

Tiankui Wang¹, Chunlei Ji*, Yuanfeng Hao² and Jianli He³

¹ School of Electronic Information, Shanghai Dianji University, Shanghai, 201306, China
² School of Electronic Information, Shanghai Dianji University, Shanghai, 201306, China
³ School of Electronic Information, Shanghai Dianji University, Shanghai, 201306, China
*Corresponding author’s e-mail: jicl@sdju.edu.cn

Abstract. Aiming at the multi-objective problem of flow workshop problem, a multi-objective optimization model was constructed and an improved non-dominated sorting genetic algorithm was proposed. Firstly, aiming at these problems, this paper proposes a two-stage chromosome coding method to adapt to the new production scenarios. Secondly, a new adaptive method is proposed to improve the convergence speed and the superiority of Pareto solution set. Finally, simulation results show that the optimality of the improved non-dominated sorting genetic algorithm is improved greatly.

1. Introduction

Shop scheduling is no longer a traditional job shop scheduling problem (JSP) with the development of industrial informatization. Because the traditional workshop production scheduling is a predetermined process, working hours and production equipment, the efficiency is not very high. In 1990, Bruker et al. [1] proposed the Flexible Job Shop Scheduling problem (FJSP) to optimize shop floor scheduling and improve production efficiency.

Flexible shop scheduling is developed on a traditional basis. The methods used for flexible shop scheduling include taboo search (TS), simulated annealing (SA), Genetic algorithm (GA) and particle swarm optimization (PSO) [2,3,4,5]. The application effect of genetic algorithms in flexible shop scheduling is obvious, and it can solve problems quickly and get the optimal solution. Zhang guohui et al. [6] improved the population initialization method of genetic algorithm to overcome the slow local search speed and random search speed of standard genetic algorithm [7,8].

This paper presents an improved genetic algorithm, the improved non-dominated sorting genetic algorithm, to solve the problem of different batches and different quantities of work, the total completion time is minimum.

2. Materials and Methods

2.1. Problem description

The flow shop scheduling problem is usually described as n (W₁, W₂, · · ·, Wₙ) products are produced on m (M₁, M₂, · · ·, Mₘ) machines. Each product has p processes and each piece of equipment is assigned
different processes for production. The j-th process of the i-th product is represented as $W_{ij}$, and the p process of n products has the same processing path. It is assumed that each product is produced in the order of 1~ m, the product production order is $(W_1, W_2, \cdots, W_n)$, and the time of producing product $w_i$ on machine g. Given the maximum production efficiency, reasonable resource allocation, and minimum inventory of flow shop, and in line with the time expectation of users on product orders, this paper considers the maximum completion time $C_{\text{max}}$ and total process time $C_{\text{total}}$ as the minimum. Among them:

(1) The corresponding objective function of maximum completion time:

$$C_{\text{max}} = \max (C_{W,m} | i \in 1, 2, \cdots, n)$$  \hspace{1cm} (1)

(2) The objective function of the total process energy consumption:

$$C_{\text{total}} = \sum_{i=1}^{n} C_{w,m}$$  \hspace{1cm} (2)

2.2. Algorithm solution process

First, the parent population is generated through two-stage coding, and the parent population is obtained after the population is classified. By identifying non-dominated individuals and assigning appropriate virtual fitness values, all individuals in the population are gradually stratified. After all, individuals are stratified, adaptive crossover and mutation operations are performed on the population through the set adaptive operator. Until the population reaches the maximum number of iterations. The flowchart of the algorithm is shown in Figure 1.

![Figure 1. Algorithm flowchart](Image)
2.3. Two-segment chromosome encoding
The chromosome encoding method selects integer encoding, the encoding is divided into 2 segments, the number of chromosomes in each segment is the same, the first segment corresponds to the first half of the chromosome, indicating the processing order of the workpiece on the machine; the second segment corresponds to the second half of the chromosome, indicating the process selects the number of the processing machine. For example, the chromosomes of 5 jobs and 3 machines are shown in Figure 2 below.

![Figure 2. Chromosome composition](image)

2.3.1. Adaptive crossover and mutation probability
Randomly assign two weighting factors $\alpha$ and $\gamma$, and set the crossover probability $P_c$ and mutation probability $P_m$ that are both affected by the number of iterations and fitness value, and the calculation is as follows:

$$P_c = \begin{cases} P_{c,\text{max}} - (P_{c,\text{max}} - P_{c,\text{min}}) \left(1 - \alpha \right) \left(\frac{f' - f_{\text{avg}}}{f_{\text{max}} - f_{\text{avg}}} \right), & f' > f_{\text{avg}} \\ P_{c,\text{max}}, & f' \leq f_{\text{avg}} \end{cases}$$ (3)

$$P_m = \begin{cases} P_{m,\text{max}} - (P_{m,\text{max}} - P_{m,\text{min}}) \left(1 - \gamma \right) \left(\frac{f' - f_{\text{avg}}}{f_{\text{max}} - f_{\text{avg}}} \right), & f' > f_{\text{avg}} \\ P_{m,\text{max}}, & f' \leq f_{\text{avg}} \end{cases}$$ (4)

In the above formula, $P_{c,\text{max}}$ and $P_{m,\text{max}}$ are the maximum crossover probability and the maximum mutation probability. $P_{c,\text{min}}$ and $P_{m,\text{min}}$ are the minimum crossover probability and the minimum mutation probability. $f'$ is the fitness value of the mutated individual, $f_{\text{max}}$ is the maximum fitness value of the individuals in the population, $f_{\text{avg}}$ is the average fitness value of the individuals in the population.

3. Results & Discussion
In order to verify the effectiveness of the algorithm, the traditional non-dominated sorting genetic algorithm and the improved non-dominated sorting genetic algorithm are run 100 times each, every 20 times as a group, a total of 5 groups. The maximum, minimum, and average values of each optimal solution set are calculated through weight distribution. The final results are shown in Table 1.
Table 1. Comparison of experimental results before and after improvement

| Experiment group number | Algorithm group       | Maximum | Minimum | Average value |
|-------------------------|-----------------------|---------|---------|---------------|
| 1                       | Before improvement    | 40.1    | 32.4    | 35.18         |
|                         | After improvement     | 39.8    | 29.8    | 34.3          |
| 2                       | Before improvement    | 41.2    | 33.5    | 36.4          |
|                         | After improvement     | 38.6    | 28.1    | 36.9          |
| 3                       | Before improvement    | 38.7    | 26.8    | 34.1          |
|                         | After improvement     | 39.1    | 26.4    | 32.75         |
| 4                       | Before improvement    | 44.5    | 32.1    | 39.3          |
|                         | After improvement     | 43.1    | 30.1    | 37.9          |
| 5                       | Before improvement    | 44.8    | 32.4    | 39.5          |
|                         | After improvement     | 40.1    | 34.1    | 38.1          |

Randomly select a comparison group for comparison. The comparison of the obtained frontier surface diagrams is shown in Figure 3:

![Figure 3. Pareto front face comparison before and after improvement](image-url)
After the above comparative experiments, it can be proved that the improved non-dominated sorting genetic algorithm is better than the traditional NSGA algorithm in the span of the solution set or the distribution surface of the optimal solution set Pareto optimal solution set. Characteristics.

4. Conclusion
Green production is the inherent requirement of building a manufacturing power. Based on the NSGA algorithm, this paper proposes a two-stage coding method and an adaptive crossover and mutation strategy based on the two goals of minimizing energy consumption and minimizing completion time. Simulation experiments show that the improved NSGA algorithm has good results in terms of solution breadth and quality, and has strong practical significance.

Acknowledgments
This article is one of the phased achievements of the National Natural Science Youth Fund Project 《Research on Online Query Technology of Dynamic Graphs Based on Distributed Stream Processing》 (61702320).

References
[1] Bruker P, Schlic R. Job-shop scheduling with multi-purpose machines[J]. Computing, 1990,4(2):369-375.
[2] Xiao Huajun. Research on dual resource flexible job shop scheduling problem considering energy efficiency[D]. Wuhan: Huazhong University of Science and Technology, 2018.
[3] Liu Yi. Research on workshop dynamic scheduling based on simulated annealing genetic algorithm[D]. Jinan: Shandong University, 2017.
[4] Zhang Chaoyong, Rao Yunqing, Li Peigen, et al. Two-level genetic algorithm for flexible job shop scheduling problem[J]. Journal of Mechanical Engineering, 2007, 43(4):119-124.
[5] Liu Min, Zhang Chaoyong, Zhang Guojun, et al. Research on displacement flow shop scheduling problem based on hybrid particle swarm optimization algorithm[J]. China Mechanical Engineering, 2011, 22(17):2048-2053.
[6] Zhang Guohui, Gao Liang, Li Peigeng, et al. Improved genetic algorithm for flexible Job shop scheduling problem[J]. Journal of Mechanical Engineering, 2009, 45(7):145-151.
[7] Yu Lu. Research on flexible Manufacturing System scheduling based on Hybrid Genetic Algorithm[D]. Zhenjiang: Jiangsu University, 2016.
[8] Liu Jianye, Wang hua, Zhou Wanmeng. Low orbit constellation sensor resource scheduling algorithm based on GA-SA[J]. China Mechanical Engineering, 2004, 15(23):2149-2153.