An improved genetic algorithm for solving flexible job shop

Ruifeng Yang*
School of management Tianjin University of Technology, Tianjin, China
*Corresponding author: ruifengyang@tjut.edu.cn

Abstract. According to the characteristics of flexible job shop scheduling (FJPS), a mathematical model was established to minimize the maximum completion time, and an improved genetic algorithm was proposed to solve the problem. A variety of heuristic methods are used to improve the quality of the initial solution. The parallel double-chain encoding is designed and the optimal insertion method is proposed to improve the quality of the solution. Two crossover methods, namely IPOX crossover and multi-point crossover, are adopted to inherit the excellent genes from the parent generation and balance the global development ability of the algorithm. In different coding layers, a variety of variation methods were used to maintain the diversity of the population. The local development ability of the algorithm is enhanced by variable neighborhood search. Finally, by solving the Brandimarte standard example and comparing with other algorithms, the feasibility and effectiveness of the proposed algorithm are verified.

1. Introduction
Flexible job shop scheduling problem, FJSP is a kind of job shop scheduling problem. As an extension of traditional shop scheduling problem, it has a wider application environment and a more complex solving space. JSPs on more than three machines have been proven to be NP problems [1], so FJSPs as an extension problem are NP difficult as well. The FJSP problem contains two sub-problems: machine selection and procedure ordering. The two decision Spaces increase the difficulty of solving the problem. For many years, many intelligent algorithms have been used to solve shop shop scheduling problems, such as tabu search [2], genetic algorithm [3], particle swarm optimization algorithm [4], etc. However, most of these algorithms start from the algorithm itself or model without heuristic rules, which inevitably leads to algorithm performance redundancy. On the other hand, how to jump out of the local optimal solution has always been the focus and difficulty of solving FJSP and there is a lot of room for improvement. According to the above problem, this paper proposes an improved genetic algorithm new improved genetic algorithm, NIGA FJSP problem was solved.

2. Algorithm design
Genetic algorithm is an intelligent algorithm based on natural selection and genetic principles proposed by Holland in 1970 [5]. It simulates the natural environment, carries on the selection, the crossover and the mutation operation to the population, obtains the next generation population, after several generations of evolution obtains the final result. Because of its good robustness, implicit parallelism and strong global search ability, it is widely used in various fields. According to the characteristics of the FJSP model and genetic algorithm, this paper proposes the NIGA algorithm to solve the FJSP problem.
2.1. Encoding and decoding
Considering the two sub-problems of FJSP, operation sequence and machine selection, the parallel double-chain coding is adopted. The first layer is operation sequence (OS) and the second layer is machine selection coding (MS). As shown in Figure 1, the number of OS layer represents the job number, and the frequency of occurrence is the process number of the current job. The MS layer is arranged according to the working sequence of the job. The 4 in this figure indicates that the first working procedure of the No.3 job is processed on the No.4 machine. The length of OS layer and MS layer is the same, and both are PS. This coding method ensures that the solution generated by the following strategies, such as crossover, mutation and local search, is still feasible, and it is simple and flexible without any requirements on the length and quantity of the work piece. On the other hand, separate operations on one layer do not affect the other layer, with strong parallelism performance.

![Figure 1. Coding diagram](image)

The main goal of decoding is to obtain a high-quality solution in the spatial range according to the form of coding layer. This paper proposes an optimal insertion method, in order to realize the chromosome decoding to efficiently search the solution space.

1) Determining whether it is the first procedure of the job, if it is the first procedure, the idle starting point is judged to be 0, otherwise, the idle starting point is the completion time of the previous procedure of the job;
2) Find the idle period after the idle starting point, and select the time period which is greater than or equal to the processing time of the process to be processed. If there is no idle insertion that satisfies the condition, it is processed normally in order.
3) Select the idle insert that is closest to the processing time of the process to be processed;
4) Repeat 1) to 3) until all procedures are completed;
5) Calculate the maximum completion time.

2.2. Population initialization
In genetic algorithm, population initialization is the key step, and the quality of initial solution directly affects the distribution degree of population in the whole search space and the convergence speed of the algorithm. Existing studies usually adopt random initialization method, which makes the quality of the solution low and often requires more iterations to obtain better population quality.

In the single objective part of this paper, the GLS initialization method proposed in reference is adopted. The method divides the initialization of MS coding layer into three parts: global selection (GS), local selection (LS) and random selection (RS). OS coding layer is generated in a completely random way.

2.3. Crossover operation
The purpose of crossover is to generate new individuals after a series of operations, and search the solution space efficiently on the premise of keeping good genes as far as possible. Therefore, crossover operations must meet the characteristics of feasibility, inheritance, non-redundancy of information and so on. In view of the characteristics of coding mode and genetic algorithm, this paper adopts two crossover modes: IPOX crossover for OS layer chromosomes and multi-point crossover for MS layer chromosomes. Since the IPOX crossover has the characteristics of low constraint and good gene inheritance, it is used to complete the crossover operation of OS layer. MS layer adopts the sequential
coding method, which requires a high level of crossover, so it adopts the multi-point crossover method that does not destroy the effective sequence of genes.

The multi-point crossover steps are as follows:
1) randomly select two parents father1 and father2;
2) randomly generate a group of 0 and 1 arrays l with dimensions consistent with MS chromosome;
3) child individual child1 inherits the gene whose l array is 1 corresponding to father1, and child individual child2 inherits the gene whose l array is 1 corresponding to father2;
4) insert the unselected genes in father2 into the blank positions of child1 in sequence, and insert the unselected genes in father1 into the blank positions of child2 in sequence.

2.4. Multiple mutation operation
Mutation operation refers to the formation of new individuals through small amplitude of chromosome perturbation to maintain population diversity, which affects the local search ability of genetic algorithm to a certain extent. Commonly used mutation operations are interchange, reverse order, insert, etc., but they can not achieve ideal results for FJSP problems. In FJSP, machine selection often has a greater impact on the results than process arrangement, so this paper adopts multiple variation strategies of multiple machines -- random machine and minimum processing time machine selection to maintain population diversity.

2.5. Variable neighbor search policy
Although the mutation operation based on MS chromosome was introduced, which could increase the diversity of the population to some extent, the OS chromosome part was not improved, and the algorithm may still fall into the local optimum. To solve this problem, a variable neighbor search strategy for OS coding is designed, which includes three operations.

A: Geometrically arrange all the conditions of A chromosome segment.
The steps are as follows:
1) randomly select an OS chromosome;
2) select a number of pieces from n pieces at random;
3) Randomly select a procedure from each selected job;
4) The unselected OS chromosome genes were kept in situ, and the selected part was enumerated and sequenced to get a new OS chromosome;
5) Calculate the fitness of the newly obtained OS chromosome. If there is a new individual whose fitness is greater than the original OS chromosome, replace the original solution; Otherwise, do nothing.
B: A segment of OS chromosome was randomly shuffled into order.
C: Random exchange of elements at two positions in the OS coding layer
A, B and C3 methods gradually reduce the disturbance of the OS coding layer to deal with the situation of excessive convergence in the iterative process. At the initial stage of the population, strategy A with small disturbance was adopted. With the progress of the iteration, two strategies B and C were adopted to maintain the diversity of the population in the later stage.

2.6. Algorithm process
According to the above improvement strategy, this paper proposes the following solving steps of NIGA algorithm to solve the flexible job shop problem:
1) Initialize the population and set relevant parameters according to the rules proposed in 2.2;
2) The fitness of the population was calculated by OIM;
3) Select the group to be followed through the tournament selection;
4) Cross the OS layer of the selected group with IPOX, and cross the MS layer with multiple points;
5) perform the multiple mutation operation proposed in 2.4 on MS layer;
6) perform the local search operation proposed in OS layer 2.5;
7) the fitness of the new generation population was calculated and sorted in descending order according to the fitness, so as to preserve the optimal individuals of this generation;
8) The loop terminates the judgment. If it is satisfied, skip to 9), otherwise, skip to 3);
9) The optimal solution output, algorithm end.

3. Simulation experiment
The parameters are set as follows: population \( pop = 2^*n^*m \); Crossover probability \( p_c = 0.8 \); mutation probability \( p_m = 0.05 \); maximum number of iterations \( gen = 100 \); the probability of multiple mutation A, B and C operations is \( \text{iter}/\text{gen;iter}/(2^*\text{gen};\text{iter}/(2^*\text{gen})) \). Where \( \text{iter} \) is the current iteration number.

In order to verify the feasibility and effectiveness of NIGA, this paper conducts testing and analysis from two aspects: single target and multi-target. The single goal is to minimize the maximum completion time, and the initial rule ratio is 0.6:0.3:0.1. Ten numerical examples in Brandimarte were adopted to carry out a comparative test with the literature [6-9].

![MK04 convergence curve](image)

**Figure 2.** MK04 convergence curve

| Numerical example | GENACE | DPSO | HGWO | Heuristic | NIGA |
|-------------------|--------|------|------|-----------|------|
| 1                 | 40     | 41   | 40   | 44        | 40   |
| 2                 | 29     | 26   | 29   | 28        | 28   |
| 3                 | N/A    | 204  | 207  | 204       | 204  |
| 4                 | 67     | 65   | 65   | 75        | 65   |
| 5                 | 176    | 175  | 175  | 179       | 174  |
| 6                 | 67     | 79   | 79   | 69        | 65   |
| 7                 | 147    | 149  | 149  | 149       | 145  |
| 8                 | 523    | 523  | 523  | 555       | 523  |
| 9                 | 320    | 325  | 325  | 342       | 310  |
| 10                | 229    | 253  | 253  | 242       | 219  |

**Table 1.** Algorithm comparison results.
It can be seen from the running results in Table 1 that the algorithm proposed in this paper achieves the optimal solution among the four algorithms except for MK02 when solving the Brandimarte series of examples, which proves that Nniga has good performance when solving the problems of different machines and different jobs. Taking MK04 as an example, its convergence curve is shown in Fig. 2. On the whole, the solution of each generation decreases significantly, especially before the 20th generation, which indicates that the initialization strategy adopted in this paper can effectively accelerate the convergence rate and find the optimal solution around the 60th generation and keep it stable. On the other hand, although the mean value of the population also has an obvious downward trend, there is still a large gap between it and the contemporary optimal solution, which indicates that the variable neighborhood strategy adopted by NIGA algorithm effectively maintains the diversity of the population.

4. Conclusion
To solve the flexible job shop scheduling problem, a mathematical model was established to minimize the maximum completion time, the total machine load and the maximum machine load, and an improved genetic algorithm was proposed to solve it. The initial solution can be mixed in many ways to improve the quality of the initial population. The decoding method of OIM insertion is used to realize the efficient search of the understanding space and accelerate the convergence speed. Different strategies are used for different coding layers to maintain the diversity of the population, so that the algorithm can avoid falling into local optimum and the variable neighborhood search is adopted to enhance the local search ability. A Brandimarte example is run to verify the effectiveness and optimization performance of the proposed algorithm from a single target. In order to respond to the national call and the demand of actual production, the multi-plant and multi-objective distributed scheduling with efficiency and carbon emission as the target will be studied in the future.

References
[1] BUCKER P,SCHLIE R. Job-shop scheduling with multi-purpose machines[J]. Computation,1990,4(45): 369-375.
[2] LI J,PAN Q,XIE S. A Hybrid Pareto-Based Tabu Search for Multi-objective Flexible Job Shop Scheduling Problem with E/T Penalty [C]// Advances in Swarm Intelligence. Lecture Notes in Computer Science. Berlin: ICSI,2010: 620-627..
[3] Lin Bi, Xie Minghong. Realization of Genetic Algorithm for Jobshop Scheduling Problem [J].Journal of Chongqing Institute of Technology (Natural Science Edition), 2008(6) : 73-78.
[4] SUNDAR S,SUGANTHAN P N,CHUA T J. A Swarm Intelligence Approach to Flexible Job Shop Scheduling Problem with No-Wait Constraint in Remanufacturing [C]// Artificial Intelligence and Soft Computing. Lecture Notes in Computer Science. Berlin: ICAISC,2013: 593-602..
[5] HOLLAND J H. Adaptation in natural and artificial systems[M]. Ann Arbor: The University of Michigan Press,1992: 126-137.
[6] HO N B,TAY J C. GENACE: An efficient cultural algorithm for solving the flexible job shop problem [C]// Proceeding of IEEE Congress on Evolutionary Computation. Portland: CEC,2004: 1759-1766.
[7] MOHSEN Z. A heuristic algorithm for solving flexible job shop scheduling problem[J]. The International Journal of Advanced Manufacturing Technology,2014,71(1/4): 519-528.
[8] NOUIRI M,BEKRAR A,JEIMAI A,et al. An effective and distributed particle swarm optimization algorithm for flexible job shop scheduling problem[J]. Journal of Intelligent Manufacturing,2018,29(3): 603-615.
[9] Jiang Tianhua. Hybrid Grey Wolf Optimization Algorithm for Flexible Job-shop Scheduling Problem [J]. Control and Decision, 2018,33 (3) : 503-508