

An Improved Genetic Algorithms for Multi-objective Hybrid Flow-shop Scheduling Problem

Zhao Zijin¹, Wang aimin¹, Ge Yan¹ and Lin Lin²

¹Department of Mechanical Engineering, Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing, China
²The six branch factory, Beijing Xinghang Mechanical and Electrical Equipment Co., Ltd., No. 9, DongwangzooBei Road, Fengtai District, Beijing, China

Abstract. To deal with the multi-objective hybrid flow Shop Scheduling Problem (HFSP), an improved genetic algorithms based on parallel sequential moving and variable mutation rate is proposed. Compared with the traditional GA, the algorithm proposed in this paper uses the two-point mutation rule based on VMR to find the global optimum which can make the algorithm jump out of the local optimum as far as possible, once it falls into the local optimum quickly. Decoding rules based on parallel sequential movement ensures that the artifact can start processing in time, so that the buffer between stages in the flow-shop is as little as possible, and the production cycle is shortened. Finally, a program was developed with the actual data of a workshop to verify the feasibility and effectiveness of the algorithm. The result shows that the algorithm achieves satisfactory results in all indexes mentioned above.

1 Introduction

The Hybrid Flow Shop Scheduling Problem (HFSP) [1-4] is one of the important branches of production scheduling and combinatorial optimization problems. In order to make all tasks, with multi-product and change demand, completed as early as possible before their due date, some enterprises often have the demand of minimizing the maximum completion time and the minimum penalty for the delay of all tasks.

In recent years, numerous national and international research studies on HFSP have been conducted. NENAD [5] proposed a variable neighborhood search algorithm, which extends the search range by changing the neighborhood structure set several times and obtains the local optimal solution. LOW.C[6] proposed a simulated annealing algorithm to solve the HFSP problem of unrelated parallel computers. Alaykyran K[7] used Ant Colony Algorithm to solve hybrid flow shop scheduling problem. Wang Shengyao[8] proposed a distribution estimation algorithm based on permutation encoding and decoding method for HFSP.

The above researches show that HFSP can be solved by different kinds of intelligent algorithms, and local optimal solutions can be obtained. However, genetic algorithms are apt to fall into local optimum[9] and none of them can solve the problem with both the goals of minimizing the maximum completion time and the penalty for tasks delay at the same time.

2 Problem Description and Mathematical Modeling

2.1 Problem Description.

Multi-objective HFSP can be described as: Allocating n tasks with multi-product and change demand to S stages in flow-shop, in which each stage has at least one optional machines and at least one stage has parallel machines by reasonably planning the order of tasks to minimize the maximum completion time (C_{\max}) and the penalty for total delay (D_{total}).

To better solve this problem, some assumptions can be put forward:

- (1) A single stage of a task share the same processing time on different machines;
- (2) All tasks have no priority and can be processed at time 0;
- (3) All machines share the same processing day system.

2.2 Symbol Descriptions & Mathematical Modeling.

Parameters:

O_i : task i , $i = 1, 2, \dots, n$;

N_i : size of task i ;

M^j : the Set of parallel machines of stage j ,
 $j = 1, 2, \dots, m$, in which m is the count of stages;

M_k^j : Machine k in stage j , $k = 1, 2, \dots, s_j$,

s_j is the number of parallel computers in stage j ;

$T_{i,j}$: Processing time of the j -th operation of O_i ;

- C_i :Completion time of task i ;
- D_i :Delivery of task i ;
- P_i equals 1 if C_i is later than D_i , otherwise 0;

Decision variables:

$X_{i,k,j}$ equals 1 if the j -th process of task i is arranged on M_k^j , otherwise 0;

$s_{i,j}$:Time to start processing in the j -th process of task i ;

$e_{i,j}$: The end time of the j -th process of order i ;

Based on the aforementioned assumptions and notations, the model is presented as follows:

$$\min(C_{\max} = \max\{C_1, C_2, \dots, C_n\}). \quad (1)$$

$$\min\left(D_{\text{total}} = \sum_{i=1}^n (C_i - D_i) \cdot P_i\right). \quad (2)$$

$$f = \min(C_{\max}) + \min(D_{\text{total}}). \quad (3)$$

Subject to:

$$C_i = s_{i,m} + T_{i,m} \cdot N_i, \forall i. \quad (4)$$

$$e_{i,j} = s_{i,j} + T_{i,j} \cdot N_i. \quad (5)$$

$$s_{i,(j+1)} \geq \begin{cases} s_{i,j} + T_{i,j} & , T_{i,(j+1)} > T_{i,j} \\ e_{i,j} - T_{i,(j+1)} \cdot (N_i - 1) & , T_{i,(j+1)} \leq T_{i,j} \end{cases} \quad (6)$$

$$(s_{i,j} - e_{p,q}) \cdot (s_{p,q} - e_{i,j}) \cdot (X_{i,j,k} + X_{p,q,k}) \leq 0 \quad (7)$$

$$, \forall i, j, p, q, k .$$

Equation (1) - (3) is the objective of the problem, representing the minimization of the maximum completion time, the minimization of task delay penalty and the objective function. Equation (4) - (7) is the constraint of this problem. Equation (4) denotes that the completion time of each task is equal to the largest processing completion time of the last stages. Equation (5) denotes that the processing completion time of each operation is equal to its processing start time plus its processing time. Equation (6) is process route constraint: To ensure No-idle flow shop scheduling, when the processing time of the latter stage is greater than that of the former one, the latter stage could not start until the former stage produce at least one artifact before it can start; otherwise, the end time of the latter stage must be at least longer than that of the former stage the processing time of a workpiece.

3 An Improved Genetic Algorithms Based on Parallel Sequential Moving and Variable Mutation Rate

An improved genetic algorithm based on parallel sequential moving and variable mutation is proposed in this paper. The algorithm mainly improves the mutation and decoding process to achieve the goal of minimizing the maximum completion time and the penalty for delay. Figure 1 is the processing flow.

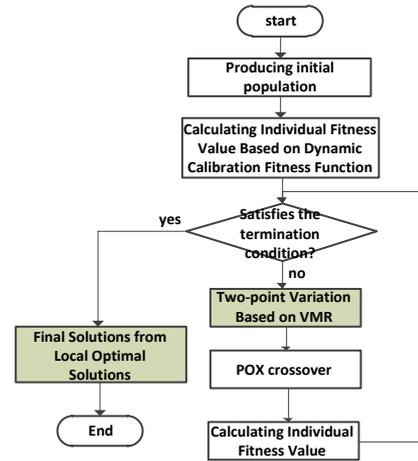


Figure 1. The processing flow of the algorithm proposed.

The algorithm proposed has the following characteristics:

(1) Using the fitness function based on dynamic calibration, the selection pressure can be dynamically adjusted with the progress of iteration in the selection process.

(2) When the algorithm falls into the local optimum quickly, the algorithm can jump out of the local optimum as far as possible by using the two-point mutation rule based on VMR to find the global optimal solution.

(3) The minimization of the delay penalty and the maximum completion time can be achieved by decoding rules based on parallel sequential movement.

3.1 Initial population generation.

For HFSP, process-based coding is generally used. Taking an example of a coding (2,1,3,1,2,3,2,3,1), in which the genes represent the processing sequence of operations in tasks, that is the first operation of task 2 - the first operation of task 1 - the first operation of task 3 - the second operation of task 1 - the second operation of task 2 - the second operation of task 3 - the third operation of task 2 - the third operation of task 3 - the third operation of task 1.

3.2 Fitness Function Based on Dynamic Calibration.

The fitness function of dynamic calibration adds ξ^k to adjust the selection pressure, i.e. the difference of selection probability between excellent and general individuals. Equation (8) - (9) makes the wide-area search interval wider to maintain the diversity of population, while the local search interval finer to maintain convergence. Adjust by adjusting the value M, r .

$$(8)$$

$$f' = f - f_{\min} + \xi^k .$$

$$\xi^0 = M, \quad \xi^k = \xi^{k-1} * r, \quad r \in [0.9, 0.999] . \quad (9)$$

3.3 VMR Two-point Variation Rule Based on HFSP.

In this paper, a two-point mutation rule of VMR (Variable Mutation Rate) based on HFSP is proposed. At the beginning of iteration, random global two-point mutation is carried out among the whole chromosome gene, but the mutation rate of the gene in the anterior segment of chromosome is higher than that in the posterior segment of chromosome. As the iteration proceeds, the mutation rate of the anterior segment of chromosome decreases and that of the posterior segment of chromosome increases. Such mutation rule can make chromosomes effectively jump out of local optimum and seek global optimal solution as far as possible.

3.4 Crossover Rule.

The crossover rule determines whether the offspring have good inheritance of the good genes of their parents and whether the offspring are feasible or not. Referring to various crossover operator schemes, POX crossover operator (an improved genetic algorithm for job shop scheduling) is finally selected, that is, fixing the genes corresponding to selected tasks in one parent's whole genes, and inserting other genes of another parent sequentially to generate a sub individual, which is shown in Figure 2.

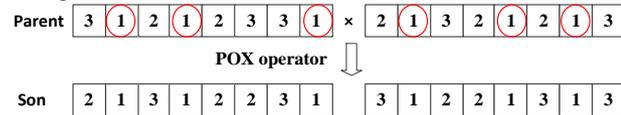


Figure 2. Crossover Rule.

3.5 Crossover Rule.

Considering the minimization of Cmax and delay, this paper adopts the decoding rule based on parallel sequential movement. The steps are as follows:

Input: Chromosome coding that currently needs to be decoded.

Output: The start time of all operations.

Step 1: Set arrangeable time of all tasks to 0, and release time of all machines to 0.

Step 2: Obtain the first gene which represents an operation that is not decoded in the current chromosome and the optional machines for the operation.

Step 3: Calculate the estimated end time and the earliest start time of the operation by $e_{i,j}^{es} = e_{i,j-1} + T_{i,j}$ and $S_{i,j} = e_{i,j}^{es} - T_{i,j} * N_i$ separately; (If the operation is the first operation of a task, then its earliest start time is 0).

Step 4: Get the release time of each machine in the set of optional machines, and compare it with the earliest start time achieved in step 2. Take the larger value as the initial arrangement time of the operation on each machine.

Step 5: Compare the initial arrangement time of the operation on the set of optional machines, take the machine which has the minimum release time to arrange the operation, and update the release time of the machine.

Step 6: Determine whether all genes on the currently imported chromosome have been decoded. If the

decoding is completed, end the process. If not, turn to step 2.

4 Experimental Study

According to the improved genetic algorithm based on the multi-objective HFSP proposed above, a simulation example has been performed in VS2010. The data of tasks and machines are collected in the actual data of a shop in Beijing. The information of the example is shown in Table 1. The selection probability of the modified genetic algorithm is 0.5, the crossover probability is 0.65. Fig.5 and Fig.6 shows the results of scheduling.

Table 1. Information of HFS.

Process work hour \ Tasks	processing time of operation 1	processing time of operation 2	processing time of operation 3
Task 1	5	4	2
Task2	2	3	2
Task3	4	2	5
Task4	2	5	2
Task5	3	2	5
Task6	4	2	3
Task7	3	3	2
Task8	4	1	5
Task9	4	2	3

processing time of operation 4	processing time of operation 5	processing time of operation 6
3	2	1
5	3	1
2	1	2
3	3	4
3	2	4
5	3	4
4	4	3
2	3	4
4	2	5

Table 2. Information of HFS.

Stages	1	2	3	4	5	6
count of parallel machines	3	3	4	3	4	4

(The device number of process 1 with three parallel computers is defined as M1-M3, and so on.)

By running the algorithm, the result is shown as a Gantt chart.

Figure 5 shows the scheduling result. As can be seen from the figure, this algorithm ensures that the operations can start processing in time, so that the buffer between stages in the flow-shop is as little as possible, and the production cycle is shortened. Therefore, this scheme should be an ideal one with less delay penalty.

As can be seen from Figure 6:

(1) The algorithm can achieve multiple convergence within a limited number of iterations and obtain multiple local optimal solutions.

(2) The algorithm can jump out of the local optimum effectively after each local optimum solution is obtained.

Although the direction is uncertain, the next local optimum solution can be quickly obtained.

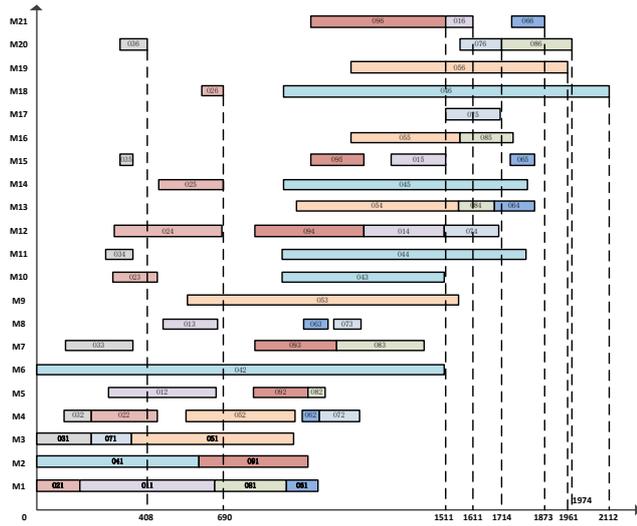


Figure 3. the final result of scheduling with Gantt chart.

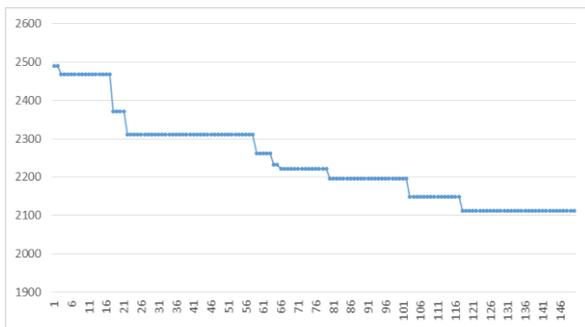


Figure 4. the result of scheduling.

5 Conclusions

(1) An Improved Genetic Algorithms Based on Parallel Sequential Moving and Variable Mutation Rate is proposed for hybrid flow shop scheduling problem with the objective of minimizing maximum completion time and total delay penalty.

(2) Compared with the traditional GA, the algorithm proposed in this paper uses the two-point mutation rule based on VMR to find the global optimum which can make the algorithm jump out of the local optimum as far as possible, once it falls into the local optimum quickly.

(3) Decoding rules based on parallel sequential movement ensures that the artifact can start processing in time, so that the buffer between stages in the flow-shop is as little as possible, and the production cycle is shortened.

References

1. Wang Ling. *Beijing: Tsinghua University Press*, 2003. 138–145
2. Cui Qi. *Computer Integrated Manufacturing Systems*, 2017, Vol.23, 1917-1928.
3. Wang Ling, Zhou Gang. *Control and Instrument in Chemical Industry*, 2011, 38(1): 1-8.
4. Portmann M C, Vignier A. *European Journal of Operational Research*, 1998, 107(2): 389–400.
5. NENAD M, PIERRE H. *Computers & Operations Research*, 1997, 24 (11): 1097-1100.
6. LOWC. *Computers and Operations Research*, 2005, 32 (8): 2013-2025I.
7. Alaykyran K. *The International Journal of Advanced Manufacturing Technology*, 2007, 35(5–6): 541–550.
8. Wang Shengyao. *ACTA AUTOMATICA SINICA*, Vol.38(2012).p. 437-443.
9. Ge Yan. *International Journal of Production Research*, vol. 3, 2017, pp. 1-17.