

# Optimizing for Blocking Flexible Flow Shop Scheduling Problem Based on Petri Net and GA

Ze Tao<sup>1, a</sup> and Qiuzhong Zhou<sup>2, b\*</sup>

<sup>1</sup>School of Mechanical Engineering, Shenyang Ligong University, Shenyang, China

<sup>2</sup>School of Automobile & Traffic, Shenyang Ligong University, Shenyang, China

<sup>a</sup>21895212@qq.com, <sup>b</sup>zqz7@sohu.com

\*The Corresponding author

**Keywords:** Flexible; Blocking; Flow shop; Petri net controller; Genetic algorithm.

**Abstract.** A Petri net with controller is used to model blocking flexible flow shop scheduling problem(BFFSP) with results obtained based on genetic algorithms(GA). The objective of scheduling problems is to minimize the maximum completing time(makes-pan). Firstly, the mathematical model is constructed for this blocking flexible flow shop scheduling problem, and computation method of make-span is obtained according to the characteristics of the BFFSP. And then a controlled Petri net model is constructed. Simulation results are obtained based on the controlled Petri net and genetic algorithm, the results show that the method proposed is feasible and efficient.

## Introduction

The blocking flow shop scheduling problems (BFSP) have been studied by many researchers[1-2]. For BFSP, there is no buffer or storage is not allowed in some stages of the manufacturing process due to technological requirements, and the completed job will stay on the machine till the next machine is available. It has been proved that the blocking flow shop scheduling problem with more than two machines is NP-hard. Therefore, it is very important both in theory and in engineering application to find effective and efficient methods for BFSP. Zhang Qiliang [3] proposed an effective population based multi-layer iterated greedy algorithm to solve the blocking flow shop scheduling problems , which objective is the make-span. In order to improve the quality of initial results, Nawaz-Enscore-Ham & Profile Fitting algorithm was proposed, and the effectiveness of the algorithm was proved through some benchmark instances. The problem of the multi-objective scheduling with make-span and total flow time minimizations for blocking flow shop is considered in [4], and a multi-objective discrete differential evolution was presented for searching alternative Pareto solution. The simulation results based on a bunch of instances show that the proposed algorithm can obtain better non-dominated solution set in Inverted Generational Distance, Set Coverage, and Hypervolume respectively. A novel quantum differential evolutionary algorithm was proposed for the BFSP to minimize the make-span in [5]. The algorithm combined Quantum Evolutionary Algorithm with Differential Evolution algorithm, and a novel quantum rotating gate was designed to control the evolutionary trend and increase the diversity of population. In order to enhance the global search ability of the algorithm and to further improve the solution quality, a Quantum-inspired Evolutionary Algorithm-Variable Neighborhood Search co-evolutionary strategy was applied.

In this paper, the blocking flexible flow shop scheduling problem (BFFSP) is studied based on Petri net controller and genetic algorithm(GA). The objective of scheduling problems is to minimize make-span. Compared to traditional flow shop scheduling problem, the unrelated parallel machines are considered in FBFSP, therefore, it is closer to practical production. In order to test the performance of the method, scheduling results based on one case is given. Comparisons and analysis between BFFSP and flexible flow shop with intermediate buffer are given.

## Mathematical Model

**Problem Description.** The blocking flexible flow shop scheduling problem can be defined as follows. There are  $n$  jobs that would be processed on  $p$  stages, there are  $M_p$  unrelated parallel machines in each stage, and each job is processed from stage 1 to stage  $p$ . In each stage, the job can be processed on any parallel machines in this stage. Since the flow shop has no intermediate buffer, the completed job stays on the machine till the next machine is available for processing. Each job must be processed on each stage, at any moment, each job can be processed on at most one machine, and each machine can process at most one job. In this paper, the objective is to plot a processing routing for each job, and to find a permutation of jobs that minimizes the maximum completion time.

**Mathematical Model.** The scheduling objective is:

$$\min Z \quad (1)$$

Let a job permutation  $s = \{s(1), s(2), \dots, s(n)\}$  represent the sequence of jobs to be processed, and  $b(s(i), j, m(j))$ ,  $i=1, \dots, n$ , denotes the beginning time of the operation  $o(s(i), j, m(j))$ ,  $j=1, \dots, p$ ,  $m(j)=1, \dots, m_j$ ,  $o(s(i), j, m(j))$  denotes job  $s(i)$  processed on the machine  $m(j)$  in stage  $j$ ,  $t(s(i), j, m(j))$  is the processing time of  $o(s(i), j, m(j))$ .

The calculation formulas of completing time for FBFSP:

$$b(s(1), 1, m(1)) = 0 \quad (2)$$

$$b(s(1), j, m(j)) = \min(b(s(1), j-1, m(j-1)) + t(s(1), j-1, m(j-1))) \quad j=2, \dots, p, \quad (3)$$

$$b(s(i), j, m(j)) = \max\{b(s(i), j-1, m(j-1)) + t(s(i), j-1, m(j-1)), \min(b(s(f), j+1, m(j+1)))\} \quad (4)$$

$f=1, \dots, i-1$ , and before processing job  $s(i)$ , job  $s(f)$  is the last job on each machine.

$$Z = \max(b(s(i), p, m(p)) + t(s(i), p, m(p)))$$

## Petri Net Controller

Design the Petri net controller based on the method proposed in [6].

In DES Petri net model, the logical conjunction of separate linear constraints involving marking only, which have the following form:

$$C = Lu \leq b \quad (5)$$

Where  $L \in Z^{n_c \times m}$ ,  $u \in Z^m$ ,  $b \in Z^{n_c}$ ,  $m$  is the number of places,  $n_c$  is the number of constraints, namely the number of controller.

1). Find the local incident matrix of constrained place based on part net design and Petri net reduction technique.

$$D = \begin{matrix} & t_1 & t_2 & \cdots & t_j \\ \begin{matrix} p_1 \\ \vdots \\ p_k \end{matrix} & \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} \\ \vdots & \vdots & \vdots & \vdots \\ d_{k1} & d_{k2} & \cdots & d_{kj} \end{bmatrix} \end{matrix} \quad (6)$$

Where  $p_1, p_2 \dots p_k$  are constrained places, namely not all  $l_{n,j}$  are zero in corresponding matrix  $L$ ,  $t_1, t_2 \dots t_j$  are transitions related to the constrained places.

2). Find the weight coefficient matrix  $L_r$  of constrained places:

$$L_r = \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1k} \\ \vdots & \vdots & \vdots & \vdots \\ l_{n,1} & l_{n,2} & \cdots & l_{n,k} \end{bmatrix} \quad (7)$$

Where  $k$  is the number of constrained places,  $L_r$  is the coefficient matrix of constrained places.

3). Find fusing places matrix  $D_c$  by Petri net reduction technique based on local incident matrix  $D$  and coefficient matrix  $L_r$  of constrained places, i.e.,

$$D_e = L_r D = \begin{matrix} & \begin{matrix} t_1 & t_2 & \cdots & t_j \end{matrix} \\ \begin{matrix} p_{e1} \\ \vdots \\ p_{en_e} \end{matrix} & \begin{bmatrix} \sum_{i=1}^k l_{1i} d_{i1} & \sum_{i=1}^k l_{1i} d_{i2} & \cdots & \sum_{i=1}^k l_{1i} d_{ij} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^k l_{ni} d_{i1} & \sum_{i=1}^k l_{ni} d_{i2} & \cdots & \sum_{i=1}^k l_{ni} d_{ij} \end{bmatrix} \end{matrix} \quad (8)$$

4). Find local incident matrix of controller places based on fusing places matrix  $D_e$ ,  $D_c = -D_e$

$$D_c = \begin{matrix} & \begin{matrix} t_1 & t_2 & \cdots & t_j \end{matrix} \\ \begin{matrix} p_{c1} \\ \vdots \\ p_{cn_c} \end{matrix} & \begin{bmatrix} -\sum_{i=1}^k l_{1i} d_{i1} & -\sum_{i=1}^k l_{1i} d_{i2} & \cdots & -\sum_{i=1}^k l_{1i} d_{ij} \\ \vdots & \vdots & \vdots & \vdots \\ -\sum_{i=1}^k l_{ni} d_{i1} & -\sum_{i=1}^k l_{ni} d_{i2} & \cdots & -\sum_{i=1}^k l_{ni} d_{ij} \end{bmatrix} \end{matrix} \quad (9)$$

Construct the controller based on matrix  $D_c$ , and its initial tokens should be:

$$u_{c0} = b - L_r u_{r0} \quad (10)$$

$u_{r0}$  is the initial token of constrained places.

### Genetic Algorithm(GA)

In theory, GA is based on probability distributions. GA realizes optimization through population generic operation based on survival of the fittest.

The details of GA are as follows:

- Create fitness function.
- Create initial population. The initial population is generated randomly, which may lead to solutions with diverse forms. If the population size is larger, although it is easier to obtain better results, the computation time is long. Therefore, it is important to select suitable population size.
- Crossover operation: Before the operation, divide population into K sub-populations: select the optimum individual in each sub-population and crossover with other individuals until producing K new populations, then select the best individual. Adopt different crossover methods in simulation, which can make the population have obvious diversity.
- Selection. Select optimum chromosome between offspring chromosome produced by crossover in different sub-population and parent. Too much selection may induce algorithm finishing prematurely, and not enough selection may induce slow convergence of the algorithm.
- Mutation: INV mutation is used to produce small perturbations on chromosomes in order to maintain the diversity of population.
- Criterion of algorithm termination: Set the eras T, stop searching if the searching eras are equal to T, and the optimal results are given.

### Case Study

In order to prove the mathematical model, Petri net controller and GA, a flexible scheduling problem is applied. There are four stages, in each stage, the quantity of parallel machines is 2. The performance of parallel machines is unrelated, and the processing information is given in table 1.

Table 1 Processing information of jobs

	Stage 1		Stage 2		Stage 3		Stage 4	
job	M1	M2	M3	M4	M5	M6	M7	M8
1	5	8	6	6	3	5	7	9
2	6	8	5	7	5	6	8	10
3	5	7	4	5	5	7	5	7
4	4	6	7	7	5	8	7	9
5	5	7	6	8	4	6	6	8
6	5	7	7	9	5	7	8	10
7	5	7	6	6	4	5	7	8
8	4	5	7	8	5	7	6	7
9	5	6	7	8	5	7	7	9
10	4	5	5	7	5	8	6	8

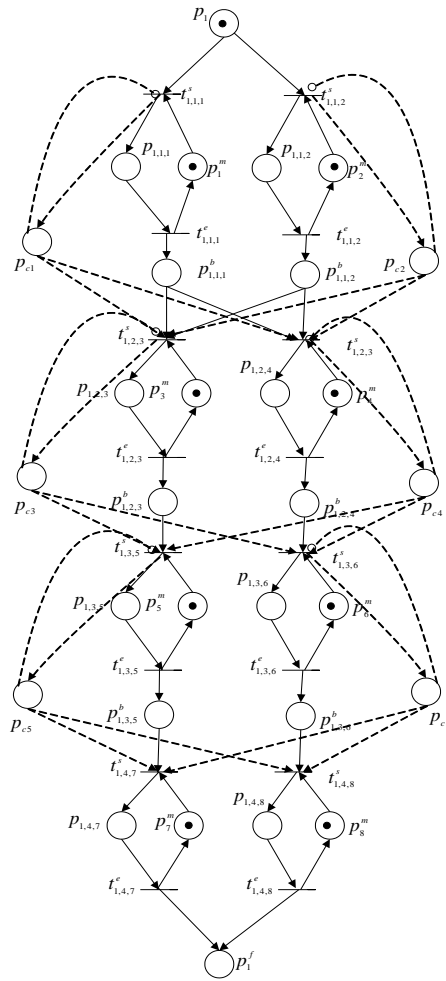


Figure 1. The Petri net model of job 1

In order to describe a Petri net model for BFSSP, a sub-Petri net is modeled, PN model of job1 as shown in Fig.1. The controller shown by the broken line in Fig. 1 controls the operation of machine. The completed job will not leave the machine till the next machine is available for processing.

$p_1$  : initial state of job 1;

$p_i^m$  : machine  $m$ ;

$p_{i,j,k}$  : operation  $j$  of job  $i$  processed on machine  $k$ ;

$t_{i,j,k}^s$  : beginning of  $p_{i,j,k}$  ;

$t_{i,j,k}^e$  : ending of  $p_{i,j,k}$  ;

$p_{i,j,k}^b$  : blocking on machine  $k$  after finishing operation  $j$  of job  $i$ ;

$p_{ci}$  : the  $i$ th controller;

$p_1^f$  : finishing of job 1.

The scheduling results are obtained based on the controlled Petri net and GA with minimizing the make-span as function objective. The Gantt graph of BFFSP is given in Fig. 2. The optimal result is 57. For machine 1, the blocking time is 3, and the job 1 does not leave the machine 1 at 19 until the next machine is available at 22 on machine 4. There is no relation of performance between the parallel machines, the machine with earlier processing time is selected.

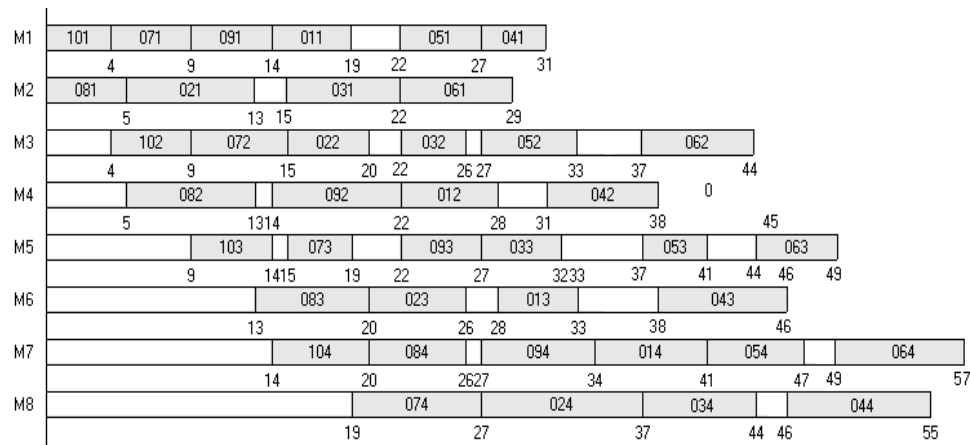


Figure 2. Gantt graph of BFFSP

## Summary

In this paper, a blocking flexible flow shop scheduling problem with unrelated parallel machines is studied based on the controlled Petri net model and GA. In the proposed method, we not only give the calculation formula of make-span, but also employ a reduced technique to design the Petri controller to control the blocking on the machine. Based on the controlled Petri net model and the computation method of make-span, the evolution of processing sequence decision can be obtained using a GA. Testing the performance of scheduling results. Moreover, the approach is shown to be effective and feasible in practical case.

## Acknowledgements

Project supported by program for Liaoning excellent talents in university (LJQ2015096).

## References

- [1] Sina Hakimzadeh Abyaneh and M. Zandieh, "Bi-objective hybrid flow shop scheduling with sequence-dependent set up times and limited buffers", *The International Journal of Advanced Manufacturing Technology*, Vol.58, 2012, pp.309-325.
- [2] Xie Zhanpeng, et al, "Blocking flow shop scheduling problem based on migrating birds optimization", *Computer Integrated Manufacturing Systems*, Vol.18, No.8, 2015, pp.2099-2107.
- [3] Zhang Qiliang and Yu Zuoming, "Population-based multi-layer iterated greedy algorithm for solving blocking flow shop scheduling problem", *Computer Integrated Manufacturing Systems*, Vol.22, No.10, 2016, pp.2315-2322.
- [4] Deng Guanlong and Tian Guangdong, "A discrete differential evolution algorithm for multi-objective scheduling in blocking flow shop", *Journal of East China University of Science and Technology*, Vol.42, No.5, 2016, pp.682-689.
- [5] Qi Xuemei, Wang Hongtao, et al, "Novel quantum differential evolutionary algorithm for blocking flow shop scheduling", *Journal of Computer Applications*, Vol.35, No.3, 2015, pp.663-667.
- [6] Ze Tao and Liu Xiao Xia. *Controller Design of Petri Net for OR-logics Linear Constraints*, *International Conference on Intelligent Networks and Intelligent Systems*, 2010, pp.225-228.