A Study of Bi-Criteria Flexible Flow Lines Scheduling Problems with Queue Time Constraints

Chun-Lung Chen

Abstract This paper considers the scheduling problems in a flexible flow line (FFL) with queue time constraints. The objective of the scheduling problems is to minimize the primary criterion which is exceeding queue time constraint times and the secondary criterion which is makespan. The problem considered in the paper is a NP-hard in a strong sense. It requires much computation time to find the optimal solution; therefore, heuristics are an acceptable practice for finding good solutions. In this paper, a meta-heuristic is proposed to solve the candidate problems. In order to evaluate the performance of the proposed heuristics, a conventional tabu search algorithm is examined for comparison purposes. The results show the proposed meta-heuristic performs effective.

Keywords Flexible flow line · Queue time constraints · Bi-criteria · Dispatching rule · Meta-heuristic

1 Introduction

This research is aimed at developing heuristic algorithms to solve bi-criteria of flexible flow lines scheduling problems with queue time constraints. The objective of the scheduling problems is to minimize the primary criterion which is exceeding queue time constraint times and the secondary criterion which is makespan. The flexible flow lines scheduling problems are perceived as NP-Hard problems, therefore we intend to develop heuristic methods to solve the problems in this study. Heuristics are developed for the primary criterion which is exceeding queue time constraint times and the secondary criterion which is makespan.

C.-L. Chen (✉)
Department of Accounting Information, Takming University of Science and Technology, Taipei, Taiwan, Republic of China
e-mail: charleschen@takming.edu.tw

Y.-K. Lin et al. (eds.), *Proceedings of the Institute of Industrial Engineers Asian Conference 2013*, DOI: 10.1007/978-981-4451-98-7_154,

Flexible flow line (FFL) problems are also known as flexible flow shop (FFS), hybrid flow shop (HFS), or flow shop with multiple processors (FSMP) problems. A typical flexible flow line production system could be defined in this way: There are *N* number of jobs to pass through *J* number of stages, and every stage may contain one or more than one machine, and the waiting area between stages is assumed to be infinite, in other words, there is no job jammed in the waiting area. All jobs have the same routing over the stages of the shop and the flow of jobs through the shop moves in one direction from the first stage to the last.

It is common to control queue time in the production process in the making of many products. For example, queue time control is carried out in the assembly process of LCD (Cell), between wet etch and furnace in the fabrication of wafer in semiconductor industry, and in the procedure of wafer bumping in IC packaging industry. The queue-time constraint is defined as the time elapsed between two processes (Scholl and Domaschke 2000; Ono et al. 2006). In order to avoid quality defects due to long queue time in the production line, therefore these factories set up a maximum queue time between production processes so that no part needs to be reworked or discarded due to exceeding the queue time limit that causes production capacity reduction and economic loss (Su 2003).

Since queue time constraint phenomenon usually exists in FFL, they can be classified as flexible flow line with queue time constraint problems. There are some studies to FFL with no-wait or limited waiting time constraint. Wang et al. (2005) presents a heuristic algorithm to solve a two-stage flexible flowshop scheduling problem with no waiting time between two sequential operations of a job and no idle time between two consecutive processed jobs on machines of the second stage. Some studies relax the no-wait constraint that job must be processed continuously without waiting time between consecutive stages. Su (2003) presented a heuristic algorithm and a mixed integer program to solve a hybrid two-stage flowshop with limited waiting constraints. Chen and Yang (2006) proposed eight mixed binary integer programming models for the open-shop, job-shop, flow-shop, and permutation flow-shop scheduling problems with limited waiting time. The above two studies are to minimize makespan. Gicque et al. (2012) proposed a discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints and considered the objective of minimizing the total weighted tardiness.

The mentioned above studies considered the jobs must be processed within every given queue time between each consecutive stages in the whole system. However, there are many situations usually occur in wafer fabrication and wafer bumping procedures, the mentioned above studies do not consider. For example, queue time constraints may be a time window set between two specified stages to prevent quality defects and the specified stages may not be consecutive. Figure 1 illustrates the physical relationship for queue time constraint between two specified stages. As shown in Fig. 1, there is a queue time among stage *k* to stage $k + b$, and the queue time is represented by $\mathrm{QT}_{\mathrm{k,k+b}}$, and among stage *k* to stage $k + b$ there is zero to b-1 stages. For convenience, the queue time constraint between two specified stages can be regarded as a queue time constraint block

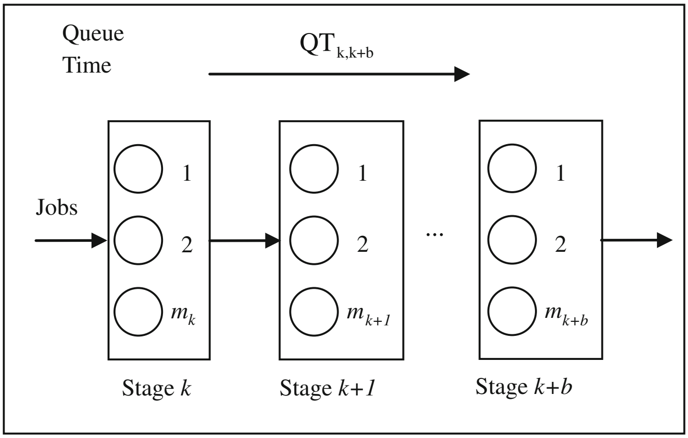

Fig. 1 Queue time constraint among stages

(QTCB). In addition, the QTCB may not include all stages in the whole FFL system. This means that there are many QTCBs in the FFL system controlled by several given queue times. For convenience, in this paper we named stage k as "the upstream specified process stage", and stage $k + 1$ as "the downstream specified process stage". In addition, we name the stages in FFL except for stages k and $k + 1$ as "general stage".

To the best of our knowledge, there are no studies that consider the bi-criteria of FFL scheduling problems with multiple QTCBs. The objective of the problem is to minimize the primary criterion which is exceeding queue time constraint times and the secondary criterion which is makespan. Hoogeveen et al. (1996) proved that minimizing makespan for two-stage FFL problems with stage 1 having one machine and stage 2 having two machines or with stage 1 having two machines and stage 2 having one machine is NP-hard. Therefore, the candidate FFL problem considered is at least NP-hard. Since it requires much computation time to find the optimal solution, heuristics are proposed to solve the candidate problems. A lot of test problems will be used to evaluate the performance of the proposed heuristics.

2 Description of the Considered Problem

The FFL with queue time constraints problems considered in this paper assumes that there are J stages and include at least a QTCB. We introduce the following characters which are considered in this paper.

There are m_j identical parallel machines in stage j and the number m_j may vary from stage to stage. Each stage has at least one machine, and at least one stage must have more than one machine. There are n jobs to be processed and each job has the same routing and must visit all the stages consecutively. Each job i has a positive processing time p_{ij}. A machine can process only one job at a time, and jobs cannot be preempted. There are unlimited buffers between stages. There are at least one QTCB existing in the system. There is no machine breakdown and all information is known in advance. If a job exceeds the queue time constraints, automatically return to the upstream specified process stage waiting re-work, and without taking into consideration the rejection. Bi-criteria are considered in the research, exceeding queue time constraint times and makespan. The objective is to find a schedule that optimize the primary criterion (exceeding queue time constraint times) followed by the optimization of the secondary criterion (makespan) subject to the primary objective value.

3 Proposed Heuristics

Due to the effect of queue time constraints, when a job is selected, it is important to pay attention to the queue time constraints lest the job to be reworked or discarded. Therefore, one needs to make several decisions and carry them out in the stage, and the first decision is whether to release a job from the waiting area to a upstream specified process stage, and secondly, to carry out the dispatching decision made under the queue time constraints at the downstream specified process stages.

The decision of upstream specified process stage is to determine whether the release of the job will collide with the queue time constraints. If it will, then the release of the job is not allowed. Due to the fact that once a job is selected at an upstream specified process stage, we have to make sure that the job won't collide with the queue time constraints at the downstream specified process stages. We develop a trial simulation method (TSM) that selects a job by the machine from the upstream specified process stage buffer, and computes the job whether it passes the queue-time constraint block.

In the downstream specified process stage, if a job stays in the waiting area for too long and exceeds the queue time limit, then it requires rework; therefore, we see the queue time limit as the most important decision criterion. In order to determine if the job exceeds the queue time limit of the various stages in the queue time area, the maximum queue time should be first established. The RQT rule is used to determine the job with the minimum remaining queue-time. The RQT can be expressed as ($\mathrm{MQT}_{\mathrm{ik}}$), where $\mathrm{MQT}_{\mathrm{ik}}$ is the maximum queue time of job i could stay in the queue at stage k.

An extended iterated local search algorithm (EILS) is developed to solve the candidate problems. The workflow of the proposed EILS algorithm is shown as Fig. 2. The figure shows that an approach can be applied to generate a feasible initial solution, and a greedy local search is applied to improve the initial solution obtained. In addition, a shaking operator, Shake(), is constructed to perturb the incumbent solution and to increase diversification of the search in EILS. In this paper, the destruct-construct method (Ruiz and Stützle 2007) is applied to generate a set of input candidate solutions which are different from the incumbent solution. Then, a minimal solution from the input candidate solutions is selected to be the input solution in the next iteration.

4 Computational Experiments and Analysis of Results

In this paper, we developed a deterministic scheduling model to verify the performance of the proposed methods and the assumptions of the model are:

(1) There are 6 stages. The routing information and the related queue time information are shown in Table 1.
(2) The number of machines at each stage is generated from the uniform distribution U(1, 5).
(3) The number of jobs is 30 and 50.

```
Procedure EILS
Generate an initial solution $S_0$.
Set $S_b = S_0$. // $S_b$ = the incumbent best solution
Set the value of exceeding queue time constraint times: $V_{qb} = f(S_0)$
Set the value of makespan: $V_{mb} = f(S_0)$
Repeat
  //apply a greedy local search and initial solution $S_0$ to obtain an improved solution $S_k$
  $S_k = LS(S_0)$
   Set $V_{qk}$ = the value of exceeding queue time constraint times obtained by LS($S_0$)
  Set $V_{mk}$ = the value of makespan by LS($S_0$)
  If $V_{qk} < V_{qb}$ Then
        $V_{qb} = V_{qk}$
        $S_b = S_k$
  ELSE IF $V_{qk} = V_{qb}$ Then
        If $V_{mk} < V_{mb}$ Then
              $V_{mb} = V_{mk}$
              $S_b = S_k$
        End If
  End If
    Apply Shake() function to generate a set of input candidate solutions which is different from the best solution
    $S_0$ = Select a minimal solution from the generated input candidate solutions
Until termination condition met
End
```

Fig. 2 The procedure of the proposed EILS algorithm

Table 1 Relevant data for job in multiple queue time constraints section

Routing number	Stage	Beginning stage with queue time constraints	Final stage with queue time constraints
N	Stage k		
N + 1	Stage $k + 1$	Stage $k + 1$	Stage $k + 3$
N + 2	Stage $k + 2$		
N + 3	Stage $k + 3$	Stage $k + 3$	Stage $k + 4$
N + 4	Stage $k + 4$		
N + 5	Stage $k + 5$		

(4) It is assumed that the processing time is generated from the uniform distribution U(10, 100) * M_j,where M_j means the number of machines at stage j.

(5) The queue time limit is generated from 100 (maximum processing time) * *q_len* * 5 (loose) and 100 * *q_len* * 4 (tight). *q_len* is the number of the downstream specified process stages within a QTBC.

(6) If a job exceeds the queue time constraints, automatically return to the upstream specified process stage waiting re-work, and without taking into consideration the rejection.

To verify the performance of the algorithms developed in this paper, we conducted computational experiments comparing the results between conventional tabu search and the proposed heuristics. The first-come first-served (FCFS) rule is used in the general and upstream specified process stages. RANDOM rule is applied to generate the input sequence of jobs. Six trials are conducted. In verifying the performance of the proposed methods, we considered the following performance measures: number of exceeding queue time constraint times, average makespan, relative deviation index (RDI), and number of best solutions (NBS). RDI is defined as:

$$RDI = \begin{cases} \frac{S_a - S_b}{S_w - S_b} & \text{if } (S_w - S_b) \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

S_a is the solution value obtained by method a, and S_b and S_w are, respectively, the best and worst values among the solutions obtained by the methods included in the comparison.

The results are presented in Table 2. In Table 2, we found that the proposed EILS with shaking 20 candidate solutions performs better than other algorithms. Furthermore, if the queue time limit is tight, the average of the RDI produced by EILS with shaking 20 candidate solutions will be 0; and the RDI of EILS with shaking one candidate solution will be 0.0776. This means that EILS with more candidate solutions performs better than EILS with one candidate solution. The proposed EILS with shaking 20 candidate solutions should be considered as the best choice for the candidate problem.

Table 2 Results for the computational experiments

Queuetime	Algorithms	Number of exceeding queue time constraint times	Average makespan	RDI	NBS
Loose	RANDOM	0	4,157	1.0000	0
	Tabu search	0	3,525	0.2528	0
	EILS with shaking one candidate solution	0	3,370	0.0363	3
	EILS with shaking 20 candidate solutions	0	3,346	0.0052	9
Tight	RANDOM	0	4,903	1.0000	0
	Tabu search	0	4,062	0.2388	0
	EILS with shaking one candidate solution	0	3,893	0.0776	0
	EILS with shaking 20 candidate solutions	0	3,810	0.0000	12

5 Conclusions

In this paper we propose a metaheuristic to solve the scheduling problems in a FFL with queue-time constraints. The objective of the scheduling problems is to minimize the primary criterion which is exceeding queue time constraint times and the secondary criterion which is makespan. To evaluate the performance of the proposed heuristic, we designed a lot of test scenarios to simulate practical shop-floor problems. For these candidate problems, we conducted computational experiments to compare the performance of the proposed heuristic. The proposed EILS with shaking 20 candidate solutions performs better than other algorithms. Our future research will consider other system characteristics, which are not included in this paper, such as machine availability and machine eligibility. Furthermore, the idea which generates a set of shaking candidate solutions by destruct-construct method can also be applied to solve other scheduling problems in the following studies.

Acknowledgments This paper was supported in part by the National Science Council, Taiwan, ROC, under the contract NSC 101-2221-E-147 -001.

References

Chen JS, Yang JS (2006) Model formulations for the machine scheduling problem with limited waiting time constraints. J Info Optim Sci 27(1):225–240

Gicquel C, Hege L, Minoux M, Canneyt W (2012) A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints. Comput Oper Res 39:629–636

Hoogeveen JA, Lenstra JK, Veltman B (1996) Preemption scheduling in a two-stage multiprocessor flowshop is NP hard. Eur J Oper Res 89:172–175

Ono A, Kitamura S, Mori K (2006) Risk based capacity planning method for semiconductor fab with queue time constraints. In: Paper presented at the 2006 IEEE international symposium on semiconductor manufacturing

Ruiz R, Stützle T (2007) A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. Eur J Oper Res 177(3):2033–2049

Scholl W, Domaschke J (2000) Implementation of modeling and simulation in semiconductor wafer fabrication with time constraints between wet etch and furnace operation. IEEE Trans Semicond Manuf 13(3):273–277

Su LH (2003) A hybrid two-stage flowshop with limited waiting time constraints. Comput Ind Eng 44:409–424

Wang ZB, Xing WX, Bai FS (2005) No-wait flexible flowshop scheduling with no-idle machines. Oper Res Lett 33:609–614