Timed Event Graph-based Scheduling for Cyclic Permutation Flow Shop

Si Cheng Ren, De Xu, Fang Wang, and Min Tan

Institute of Automation, Chinese Academy of Sciences, Beijing, China

sicheng.ren@mail.ia.ac.cn

Abstract

The cyclic permutation flow shop is one flow shop that repetitively produces product mix under permutation schedules. We propose that the cyclic permutation flow shop given the general job sequence can be modeled as a time event graph, based on which the mixed-integer programming method is applied to find the optimal schedule with maximum throughput. Two cases of cyclic permutation flow shops with different buffer capacity (i.e., with buffer and without intermediate buffer) are discussed respectively.

Keyword: timed event graph, scheduling, cyclic permutation flow shop

I. Introduction

In a flow shop, all jobs have the same route through serial machines while the processing sequence of the jobs on each machine may be different. Permutation flow shops are a special class of flow shops where the processing sequence of the jobs on each machine is identical. Garey et al. prove that the problem of scheduling for the permutation flow shop with more than 2 machines and makespan minimization as the objective is NP-complete [1]. Extensive literature has been focused on developing heuristic procedures to find sub-optimal solutions and Framinan et al. give a good review and classification of heuristics for this problem [2]. In manufacturing environment, cyclic scheduling policy is widely adopted to repetitively produce the so-called minimal part set (MPS), or product mix, where the MPS is defined as the smallest set of part of different types in proportion to a certain production requirement. The manufacturing system produces one MPS each cycle and the throughput is represented as the inverse of the cycle time. Much effort has been devoted to the study of cyclic manufacturing systems [3,4].

In this paper we deal with the problem of scheduling for cyclic permutation flow shops. The cyclic permutation flow shop given the general job sequence can be modeled as a timed event graph, based on which the mixed-integer programming (MIP) method is applied to find the optimal schedule with maximum throughput. Two cases of cyclic permutation flow shops with different buffer capacity (i.e., with buffer and without intermediate buffer) are discussed respectively. The paper is organized as follows: In section 2, we present the problem formulation and a brief introduction to Petri nets and timed event graphs. In section 3, two timed event graph models and related MIP methods are proposed. An example is presented to illustrate in section 4 and section 5 concludes the paper.

II. Problem Formulation

In a cyclic permutation flow shop (CPFS), one batch of MPS consists of n jobs, denoted by $J_1, J_2, \ldots, J_n$. All jobs are loaded by each own pallet or AGV and route in a fixed job sequence through m serial machines, denoted by $M_1, M_2, \ldots, M_m$. After being processed on all the machines, the job is unloaded from the pallet and the pallet returns immediately to pick up the job in the next batch. Under the assumption that the processing sequence of all jobs on each machine is identical, the number of possible permutation schedules or job sequences equals to $n!$. Let $\hat{J}_1 \hat{J}_2 \ldots \hat{J}_n$ denote the general job sequence and the matrix $\mathbf{\Delta}$ with entries defined as:

$$\delta_{i,k} = \begin{cases} 1 & \text{if job } k \text{ is in the position } i \text{ in the general job sequence} \\ 0 & \text{otherwise} \end{cases}$$

For a possible permutation schedule, constraints (1) and (2) should hold.

$$\sum_{i=1}^{n} \delta_{i,k} = 1, \quad k = 1,2,\ldots,n \tag{1}$$

$$\sum_{k=1}^{n} \delta_{i,k} = 1, \quad i = 1,2,\ldots,n \tag{2}$$

To simplify the analysis, the following assumptions are made:

(H1) The machine setup time and jobs transportation time are neglected.

(H2) The processing time of J_k on machine M_j, denoted by $w_{k,j}$, is deterministic and let $\mathbf{W}$ denote the processing time matrix of jobs on machines with entries $w_{k,j}$. The processing time of $\hat{J}_i$ on machine M_j, denoted by $v_{i,j}$, can be further represented as:

$$v_{i,j} = \sum_{k=1}^{n} \delta_{i,k} w_{k,j} \tag{3}$$

And rewritten as:

$$\mathbf{V}=\mathbf{\Delta W} \tag{4}$$

where $\mathbf{V}$ denote the general processing time matrix of jobs on machines with entries $v_{i,j}$.

Timed event graphs are a subclass of Petri nets [5] and defined as a 4-tuple $TEG=(P,T,F,K_0)$ where:

$P=\{p_1,p_2,\ldots,p_{|P|}\}$ is a finite set of places and each place has exactly one input transition and exactly one output transition;

$T=\{t_1,t_2,\ldots,t_{|T|}\}$ is a finite set of timed transitions;

$F \subseteq (P\times T) \bigcup (T \times P)$ is a set of directed arcs;

K_0: $P \rightarrow \{0,1,2,\ldots\}$ is the initial marking;

$P \bigcap T = \varnothing$ and $P \bigcup T \neq \varnothing$.

If places and transitions are viewed as nodes and directed arcs as directed edges, a Petri net is essentially a bipartite digraph. The transition starts firing and consumes one token in each of the upstream places after being enabled; after holding the tokens for certain time (release time) the transition ends firing and generates one token in each of the downstream places. In [6], Ramamoorthy et al. have proved that an event graph is live if and only if each circuit contains at least one token in the initial marking; the total number of tokens in each circuit is constant in all the reachable markings; if a live event graph is strongly connected, the event graph is bounded; in a live and bounded event graph, all the transitions have the same cycle time and the event graph is periodic and cycle time λ is given as:

$$\lambda = \max_{\gamma} \frac{\mu(\gamma)}{\kappa(\gamma)} \tag{5}$$

where γ denotes any circuit in the event graph; $\mu(\gamma)$ denotes the sum of release time of all the transitions in circuit ; $\kappa(\gamma)$ denotes the number of tokens circuit contains in the initial marking. Karp algorithm [7], Howard algorithm [8], linear programming method [9-11] and etc. are available for evaluating λ.

In the next section, we establish two different timed event graph models based on which MIP methods are applied to find the optimal schedule for the CPFS with throughput maximization.

III. Timed Event Graph Models of the CPFS

A. The CPFS with Buffer

The CPFS with buffer under the general job sequence $\hat{J}_1 \hat{J}_2 ... \hat{J}_n$ is modeled as the timed event graph *TEG*1 shown in Fig.1. In *TEG*1, place $M_j(\hat{J}_i)$ contains one available machine M_j(one pallet of $\hat{J}_i$) in the initial marking. It is easy to verify that each circuit is initially marked and *TEG*1 is strongly connected, which indicates that all the transitions have the same cycle time. Let $t_{i,j}$ denote the operation of $\hat{J}_i$ on M_j whose release time is $v_{i,j}$ and λ denote the cycle time. The following optimization model can be formulated to evaluate λ. Model 1:

$$\begin{aligned}
&\min \ \lambda \\
&\text{subject to} \\
&s_{i,j+1} - s_{i,j} \geq v_{i,j} \quad i = 1,2,...,n; j = 1,2,...,m-1 \\
&s_{i+1,j} - s_{i,j} \geq v_{i,j} \quad i = 1,2,...,n-1; j = 1,2,...,m \\
&s_{i,1} - s_{i,m} + \lambda \geq v_{i,m} \quad i = 1,2,...,n \\
&s_{1,j} - s_{n,j} + \lambda \geq v_{n,j} \quad j = 1,2,...,m
\end{aligned}$$

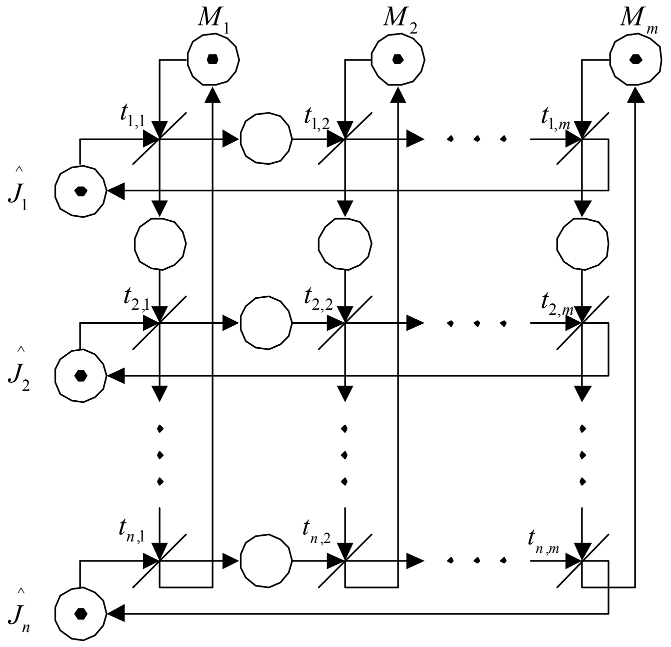

Fig.1 *TEG*1 model of the CPFS with buffer

Where $s_{i,j} \geq 0$ and denotes the time that transition $t_{i,j}$ starts firing in one cycle. The inequality constraints in Model 1 show that the inequality $s_{p\square} - s_{\square p} + K_0(p)\lambda \geq v_{\square p}$ holds for each place p with its input transition $\square p$ and output transition $p\square$. Replaced with (3) and combined with the equality constraints (1) and (2), Model 1 can be further rewritten as a MIP model in the standard form:

min λ

subject to

$$s_{i,j+1} - s_{i,j} - \sum_{k=1}^{n} \delta_{i,k} w_{k,j} \geq 0 \qquad i = 1,2,...,n;\ j = 1,2,...,m-1$$

$$s_{i+1,j} - s_{i,j} - \sum_{k=1}^{n} \delta_{i,k} w_{k,j} \geq 0 \qquad i = 1,2,...,n-1;\ j = 1,2,...,m$$

$$s_{i,1} - s_{i,m} + \lambda - \sum_{k=1}^{n} \delta_{i,k} w_{k,m} \geq 0 \quad i = 1,2,...,n$$

$$s_{1,j} - s_{n,j} + \lambda - \sum_{k=1}^{n} \delta_{n,k} w_{k,j} \geq 0 \quad j = 1,2,...,m$$

$$\sum_{i=1}^{n} \delta_{i,k} = 1 \qquad k = 1,2,...,n$$

$$\sum_{k=1}^{n} \delta_{i,k} = 1 \qquad i = 1,2,...,n$$

$$\lambda, s_{i,j} \geq 0 \text{ and } \delta_{i,k} \in \{0,1\} \tag{6}$$

where $\delta_{i,k}$ is a binary decision variable and the optimal solution $\delta^{*}_{i,k}$ is the optimal schedule for the CPFS with throughput maximization.

B. The CPFS without Intermediate Buffer

Due to the limitations of space or cost, there are cases that no intermediate buffer exists between consecutive machines. Blocking occurs when a processed job has to remain on the machine until the next machine is available. Similarly, the CPFS without intermediate buffer under the general job sequence $\hat{J}_1 \hat{J}_2 ... \hat{J}_n$ can be modeled as a timed event graph *TEG*$_2$ in Fig.2. Transition $t_{i,j}$ in *TEG*$_1$ is further decomposed into two transitions $t^{I}_{i,j}$ and $t^{O}_{i,j}$ in *TEG*$_2$, where $t^{I}_{i,j}$ denotes job $\hat{J}_i$ starts processing on M_j and $t^{O}_{i,j}$ denotes job $\hat{J}_i$ leaves M_j. The release time of transition $t^{I}_{i,j}$ is $v_{i,j}$ while that of transition $t^{O}_{i,j}$ is 0. *TEG*$_2$ is live, strongly connected and the corresponding optimization model to evaluate the cycle time is given as Model 2:

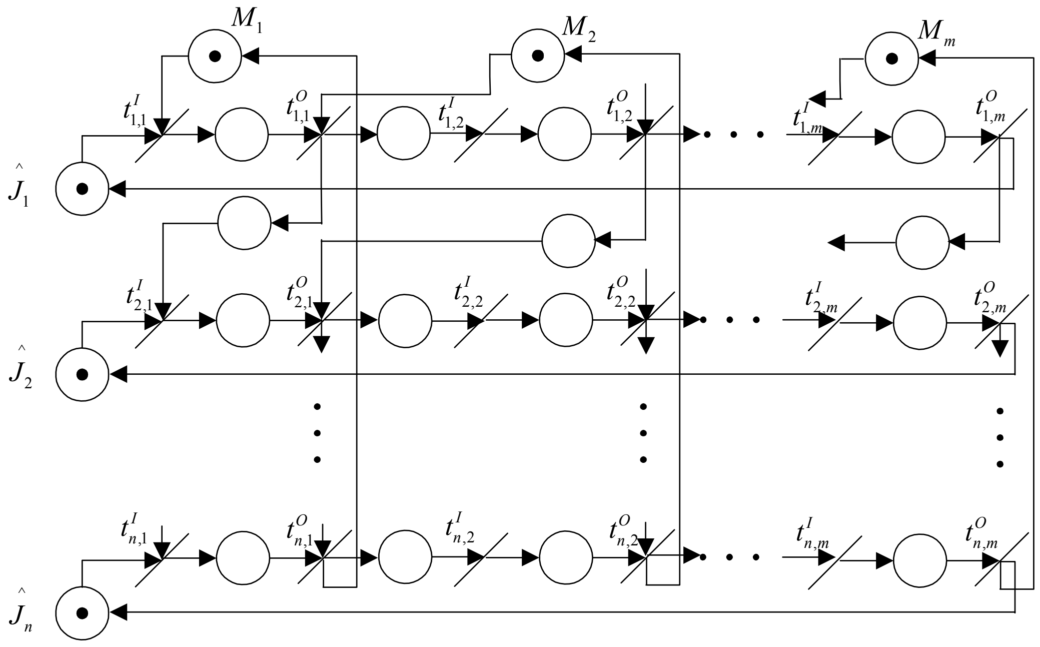

Fig.2 *TEG₂* model of the CPFS without intermediate buffer

min λ

subject to

$$s^O_{i,j} - s^I_{i,j} \geq v_{i,j} \qquad i = 1,2,...,n;\ j = 1,2,...,m$$

$$s^I_{i,j+1} - s^O_{i,j} \geq 0 \qquad i = 1,2,...,n;\ j = 1,2,...,m-1$$

$$s^I_{i,1} - s^O_{i,m} + \lambda \geq 0 \quad i = 1,2,...,n$$

$$s^I_{i+1,1} - s^O_{i,1} \geq 0 \qquad i = 1,2,...,n-1$$

$$s^I_{1,1} - s^O_{n,1} + \lambda \geq 0$$

$$s^O_{i+1,j-1} - s^O_{i,j} \geq 0 \qquad i = 1,2,...,n-1;\ j = 2,...,m$$

$$s^O_{1,j-1} - s^O_{n,j} + \lambda \geq 0\ j = 2,...,m$$

Rewrite Model 2 in the standard form:

$$\begin{aligned}
&\min \lambda \\
&\text{subject to} \\
&s_{i,j}^{O} - s_{i,j}^{I} - \sum_{k=1}^{n} \delta_{i,k} w_{k,j} \geq 0 && i = 1,2,\ldots,n;\ j = 1,2,\ldots,m \\
&s_{i,j+1}^{I} - s_{i,j}^{O} \geq 0 && i = 1,2,\ldots,n;\ j = 1,2,\ldots,m-1 \\
&s_{i,1}^{I} - s_{i,m}^{O} + \lambda \geq 0 && i = 1,2,\ldots,n \\
&s_{i+1,1}^{I} - s_{i,1}^{O} \geq 0 && i = 1,2,\ldots,n-1 \\
&s_{1,1}^{I} - s_{n,1}^{O} + \lambda \geq 0 && \\
&s_{i+1,j-1}^{O} - s_{i,j}^{O} \geq 0 && i = 1,2,\ldots,n-1;\ j = 2,\ldots,m \\
&s_{1,j-1}^{O} - s_{n,j}^{O} + \lambda \geq 0 && j = 2,\ldots,m \\
&\sum_{i=1}^{n} \delta_{i,k} = 1 && k = 1,2,\ldots,n \\
&\sum_{k=1}^{n} \delta_{i,k} = 1 && i = 1,2,\ldots,n \\
&\lambda, s_{i,j}^{I}, s_{i,j}^{O} \geq 0 \text{ and } \delta_{i,k} \in \{0,1\}
\end{aligned} \tag{7}$$

IV. Case Study

The example is selected from [12] while we further restrict that the job is not permitted to skip the machine even if there is no operation on the machine. The manufacturing system consists of three machines, M_1, M_2, M_3 in series, and one MPS consists of three jobs. The processing time of jobs on machines is given in Table 1.

Table 1. The processing time of 3 jobs on 3 machines

	M_1	M_2	M_3
J_1	0	3	4
J_2	1	2	3
J_3	5	3	0

The system cycle time under different conditions (buffer capacity or permutation schedule) is all listed in Table 2.

Table 2. The system cycle time under different conditions

Job sequences	CPFS with buffer	CPFS without buffer
$J_1J_2J_3$	9.5	11
$J_1J_3J_2$	8.5	10
$J_2J_1J_3$	8.5	10
$J_2J_3J_1$	9.5	11
$J_3J_1J_2$	9.5	11
$J_3J_2J_1$	8.5	10

The optimal solution obtained by solving Model1 (Model2) is $J_1J_3J_2$ ($J_2J_1J_3$) which is in accordance with the result in Table 2.

V. Conclusions

In this paper, we study the problem of scheduling for cyclic permutation flow shop. The CPFS under different buffer capacities are modeled as live and strongly connected time event graphs, based on which mixed-integer programs (6) and (7) are applied to find the optimal permutation schedule under which the system functions at the maximum throughput.

Acknowledgement

This work is supported by National Key Fundamental Research and Development Project of China (973, No.2002CB312200).

References

[1] M. R. Garey, D. S. Johnson, R. Sethi. The complexity of flowshop and jobShop scheduling. *Mathematics of Operations Research*. 1976, Vol.1, 117-129.

[2] J. M. Framinan, J. N. D. Gupta, R. Leisten. A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society*. 2004, Vol.55, 1243-1255.

[3] Y. Crama, J. van de Klundert. Cyclic scheduling of identical parts in a robotic cell. *Operations Research*. 1997, Vol.45, 952-965.

[4] H. Kamoun, C. Sriskandarajah. The complexity of scheduling jobs in repetitive manufacturing systems. *European Journal of Operational Research*. 1993, Vol.70, 350-364.

[5] T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*. 1989, Vol.77, 541-580.

[6] C. V. Ramamoorthy, G. S. Ho. Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Transactions on Software Engineering*. 1980, Vol.6, 440-449.

[7] R. M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*. 1978, Vol.23, 309-311.

[8] J. Cochet-Terrasson, G. Cohen, S. Gaubert, M. McGettrick, J. P. Quadrat. Numerical computation of spectral elements in max-plus algebra. in *Proc. IFAC Conferenc on System Structure and Control*, 1998.

[9] J. Campos, G. Chiola, J. M. Colom, M. Silva. Properties and performance bounds for timed marked graphs. IEEE Transactions on Circuits and Systems I. 1992, Vol.39, 386-401.

[10] J. Magott. Performance evaluation of concurrent systems using Petri nets. *Information Processing Letters*. 1984, Vol.18, 7-13.

[11] T. Yamada, S. Kataoka. On some LP problems for performance evaluation of timed marked graphs. *IEEE Transactions on Automatic Control*. 1994, Vol.39, 696-698.

[12] F. Baccelli, G. Cohen, G. J. Olsder, J. P. Quadrat. *Synchronization and Linearity: An algebra for Discrete Event Systems*. Wiley, New York, 1992.

Si Cheng Ren Ph. D. candidate in Institute of Automation, Chinese Academy of Sciences. His research interests include reconfigurable manufacturing system modeling and optimization.

De Xu Associate professor in Institute of Automation, Chinese Academy of Sciences. His research interests include robotics, computer vision and reconfigurable manufacturing system.

Fang Wang Postdoctor in Institute of Automation, Chinese Academy of Sciences. Her research interests include system reliability and advanced manufacturing systems.

Min Tan Professor in Institute of Automation, Chinese Academy of Sciences. His research interests include robotics, theory of complex system and reconfigurable manufacturing system.