A Novel Algorithm for Wafer Sojourn Time Analysis of Single-Arm Cluster Tools With Wafer Residency Time Constraints and Activity Time Variation

ChunRong Pan, Yan Qiao, NaiQi Wu, *Senior Member, IEEE*, and MengChu Zhou, *Fellow, IEEE*

***Abstract*—This paper addresses the scheduling problem of single-arm cluster tools with both wafer residency time constraints and activity time variation in semiconductor manufacturing. Based on a Petri net model developed in our previous work, polynomial algorithms are proposed to obtain the exact upper bound of the wafer sojourn time delay for the first time. With the obtained results, one can check the feasibility of a given schedule or find a feasible and optimal one if it exists. Illustrative examples are given to show the applications of the proposed method.**

***Index Terms*—Cluster tools, discrete event system, Petri net (PN), scheduling, semiconductor manufacturing.**

Nomenclature

a_i	The shortest time needed for completing a wafer at Step i, $i \in \mathbf{N}_n$.
B_i	The upper bound of the wafer sojourn time delay at Step i.
b_i	The longest time needed for completing a wafer at Step i, $i \in \mathbf{N}_n$.
I	Input function in a Petri net (PN).
K	Capacity function in a PN.
M	Marking in a PN.
$\mathbf{N} =$	$\{0, 1, 2, \ldots\}$.
$\mathbf{N}_n =$	$\{1, 2, \ldots, n\}$.
O	Output function in a PN.
PM	Process module.
PN	Petri net.
p_0	PN place modeling loadlocks.
p_i	PN place modeling Step i, $i \in \mathbf{N}_n$.
q_{i1}	PN place modeling the robot waiting before loading a wafer into Step i, $i \in \mathbf{\Omega}$.
q_{i2}	PN place modeling the scheduled robot waiting before unloading a wafer from Step i, $i \in \mathbf{\Omega}$.
q_{i3}	PN place modeling the unscheduled robot waiting before unloading a wafer from Step i, $i \in \mathbf{\Omega}$.
RCP	Real-time control policy.
r	PN place modeling the single-arm robot.
s_{i1}	PN transition modeling the robot task of loading a wafer into Step i, $i \in \mathbf{N}_n$.
s_{01}	PN transition modeling the robot task of loading a completed wafer into a loadlock.
s_{i2}	PN transition modeling the robot task of unloading a wafer from a PM at p_i and moving to p_{i+1}, $i \in \mathbf{N}_{n-1}$.
s_{02}	PN transition modeling the robot task of unloading a wafer from a loadlock at p_0 and moving to p_1.
s_{n2}	PN transition modeling the robot task of unloading a wafer from p_n and moving to a loadlock.
y_{i1}	PN transition modeling robot's moving from p_{i+2} to p_i without carrying a wafer, $i \in \mathbf{N}_{n-2} \cup \{0\}$.
$y_{(n-1)1}$	PN transition modeling robot's moving from p_0 to p_{n-1} without carrying a wafer.
y_{n1}	PN transition modeling robot's moving from p_1 to p_n without carrying a wafer.
Λ_i	Wafer sojourn time at Step i under normal conditions, $i \in \mathbf{N}_n$.
$\mathbf{\Omega} =$	$\{0\} \cup \mathbf{N}_n$.
Θ_i	Accumulated robot time delay when the robot arrives at q_{i3} for unloading a completed wafer.
$\mu_{y_{i1}}$	Time needed for firing y_{i1}, $i \in \mathbf{\Omega}$ and $\mu_{y_{i1}} \in [\alpha, \beta]$.
θ	Cycle time of the system.
θ_i	Cycle time of Step i, $i \in \mathbf{N}_n$.
κ_{id}^j	Transition firing sequence for robot's going from Steps i to d in the jth robot cycle.
λ_{i1}	Time needed for firing s_{i1}, $i \in \mathbf{\Omega}$, and $\lambda_{i1} \in [c, d]$.
λ_{i2}	Time needed for firing s_{i2}, $i \in \mathbf{N}_n$, and $\lambda_{i2} \in [c + \alpha, d + \beta]$.

Manuscript received December 30, 2013; revised August 4, 2014; accepted October 28, 2014. Date of publication December 4, 2014; date of current version April 13, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 60974098 and Grant 71361014 and in part by the Fundo para o Desenvolvimento das Ciencias e da Tecnologia of Macau under Grant 065/2013/A2. This paper was recommended by Associate Editor M. Jeng. *(Corresponding authors: MengChu Zhou and Naiqi Wu.)*

C. R. Pan is with the School of Mechanical and Electrical Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China. (e-mail: chunrongpan@163.com)

Y. Qiao is with the Department of Industrial Engineering, School of Electro-Mechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China.

N. Q. Wu is with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China, and also with the Department of Industrial Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: nqwu@must.edu.mo).

M. C. Zhou is with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China, and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982 USA (e-mail: zhou@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2014.2368995

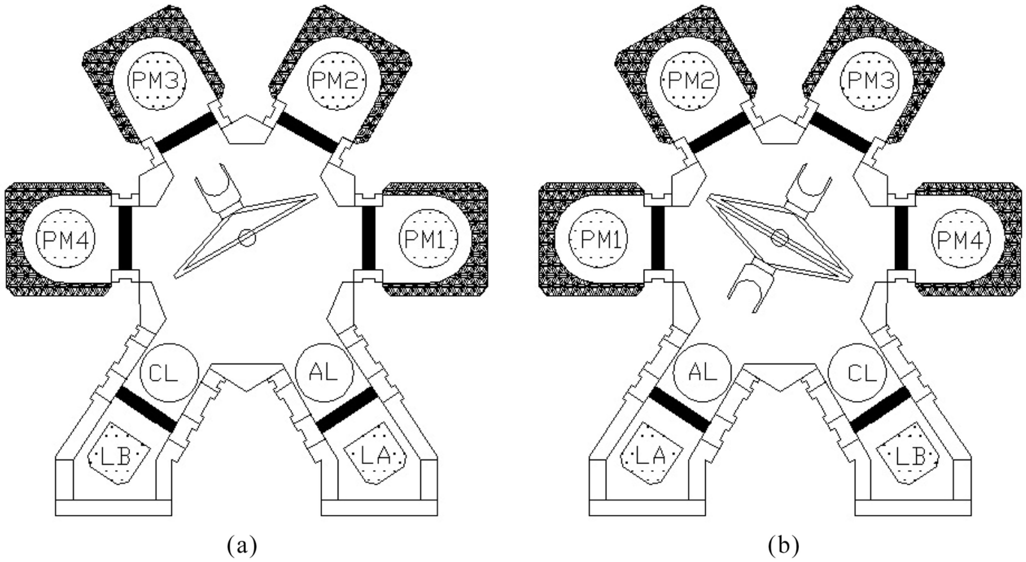

Fig. 1. Cluster tools. (a) Single-arm robot. (b) Dual-arm robot.

- λ_{02} Time needed for firing s_{02} and $\lambda_{02} \in [c_0 + \alpha, d_0 + \beta]$.
- τ_i Wafer sojourn time in Step i, $i \in \mathbf{N}_n$.
- ω_{i1} Scheduled robot waiting time before loading a wafer to Step i, $i \in \mathbf{\Omega}$.
- ω_{i2} Scheduled robot waiting time before unloading a wafer from Step i, $i \in \mathbf{\Omega}$.
- ω_{i3} Unscheduled robot waiting time before unloading a wafer from Step i, $i \in \mathbf{\Omega}$.
- δ_i The longest time for which a wafer can stay in a PM at Step i after it is processed, $i \in \mathbf{N}_n$.
- ς_i Time needed for processing a wafer at Step i, $i \in \mathbf{N}_n$, and $\varsigma_i \in [a_i, b_i]$.
- ζ^j The actual time taken for completing the jth activity.
- ψ Robot cycle time.
- ψ_1 Robot cycle time with no robot waiting.
- ψ_2 Robot waiting time in a cycle.
- $\|\bullet\|$ The exact upper bound of time delay during the execution of an activity sequence.

I. Introduction

IN SEMICONDUCTOR manufacturing, more and more manufacturers adopt cluster tools to process wafers by using single-wafer processing technology. A cluster tool consists of several process modules (PMs), an aligner, a wafer handling robot, and two loadlocks for wafer cassette loading/unloading. In general, raw wafers in a cassette have an identical recipe [1], [2]. They are loaded into a cluster tool through its loadlock, and then processed in one or more PMs with a prespecified order. After all operations are completed, they are returned to their loadlocks by the robot [3]. Such a tool can provide a flexible, reconfigurable, and efficient environment for semiconductor manufacturing, resulting in higher yield, shorter cycle time, better utilization of costly space, and lower capital cost [4]–[8]. With one or two robot arms, it is called a single and dual-arm cluster tool as shown in Fig. 1, respectively.

Extensive work has been done about modeling and analysis of cluster tools [5], [7]–[17]. It is found that, under the steady state, they operate in either the process or transport-bound region. For the former, the robot has idle time and the processing time in PMs determines the cycle time. For the latter, the robot is always busy and the cycle time is determined by its activity time. It is also shown that the PM activities follow the robot tasks [18], [19]. Hence, the key is to schedule the robot. Dispatching or priority rules are developed to do so [14], [20]. The robot moving time from one PM to another can be treated as a constant and is much shorter than the wafer processing time [1]. For single-arm cluster tools, a backward scheduling strategy is optimal [21], [22]. This is true only if there is no limit on how long a wafer can stay in PMs after it is done.

Some wafer fabrication processes pose a strict constraint on the wafer sojourn time in a PM called a wafer residency time constraint [1]–[3], [23]–[27]. With such constraints, methods for finding an optimal periodic schedule for dual-arm cluster tools are proposed in [1], [2], and [24]. Their computational efficiency is improved by deriving necessary and sufficient schedulability conditions for both single and dual-arm cluster tools as revealed in [3] and [28]. If schedulable, closed-form algorithms are given to find an optimal one.

Some wafer fabrication processes are repeated processes, or there is wafer revisiting. In [5], [7], [8], [37], and [38], scheduling strategies are presented for such tools dealing with wafer revisiting. Furthermore, an efficient technique is proposed in [39] to schedule a dual-arm cluster tool coping with both wafer revisiting and residency time constraints.

PMs in cluster tools are failure-prone. Thus, effective control policies are proposed to respond to such failures for single-arm cluster tools in [35] and [36].

All the above studies are conducted without considering activity time variation that occurs in practice. Such variation can make a feasible schedule obtained under the assumption of deterministic activity time infeasible. Methods are proposed to deal with abnormal events and activity time fluctuation in [18] and [29]. Kim and Lee [30] studied the schedulability problem for dual-arm cluster tools with bounded activity time variation. They identify so-called always schedulable and never schedulable cases by using PNs and a branching technique.

Wu and Zhou [25] show that some never schedulable cases identified in [30], in fact, are always schedulable by using their newly proposed real-time controller. By using PNs, for dual-arm cluster tools with wafer residency time constraints and activity time variation, the wafer sojourn time fluctuation is analyzed and closed-form scheduling algorithms are proposed to find an optimal schedule [25], [26], [31]. With wafer residency time constraints and activity time variation, it is much more complex to schedule single-arm cluster tools than dual-arm ones [23], [32]. Thus, by following the idea in [25] and [26], the scheduling problem of single-arm cluster tools is solved in [23], [32], and [34].

Since the time variation of both robot activities and wafer processing affects the wafer sojourn time delay in a PM in a complex way, it is very difficult to calculate the wafer sojourn time delay in a PM. In fact, the upper bound of wafer sojourn time delay in a PM obtained in [32] is not the exact one but overestimated. This makes the schedulability conditions in [23] and [34] sufficient only, but not necessary. This implies that, by the method in [23] and [34], some nonschedulable cases are, in fact, schedulable. Then, can an exact upper bound of the wafer sojourn time delay in a PM be found? If so, the schedulability conditions in [23] and [34] become necessary and sufficient conditions. This motivates us to conduct this investigation.

This work makes the following contributions: 1) the mechanism about how the activity time variation affects the wafer sojourn time is revealed; 2) algorithms are derived to calculate its exact upper bound; and 3) they are shown to be polynomial with respect to the number of parallel PMs and number of operations. Therefore, the results are significant in this research field.

The remainder of this paper is organized as follows. The next section introduces a Petri net (PN) model and real-time control policy (RCP). Then, Section III presents the algorithms for calculating the exact upper bound of wafer sojourn time delay. Illustrative examples are used to show their applications in Section IV. Finally, the conclusion is given in Section V.

II. PN Modeling and RCP

In this section, we briefly introduce the PN model developed in [32] and [33] such that this paper is self-complete.

A. PN Model for the Wafer Flow

PNs are widely used in modeling and analysis of discrete event systems [1], [17], [29], [40]–[53]. The PN model in [32] and [33] is a kind of finite capacity PN whose concept is based on [19] and [54]. It is defined as PN $= (P, T, I, O, M, K)$, where P is a finite set of places; T is a finite set of transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$; I: $P \times T \to \mathbf{N} = \{0, 1, 2, \ldots\}$ is an input function; O: $P \times T \to \mathbf{N}$ is an output function; M: $P \to \mathbf{N}$ is a marking representing the number of tokens in places with M_0 being the initial marking; and K: $P \to \mathbf{N}\backslash\{0\}$ is a capacity function, where $K(p)$ represents the largest number of tokens that p can hold.

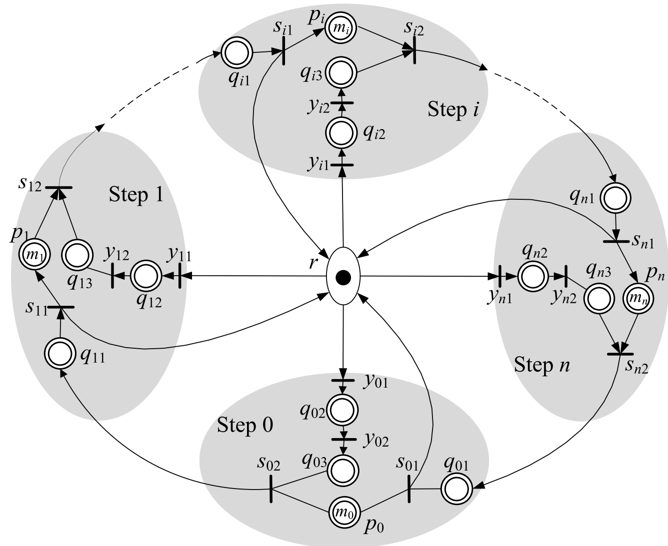

Fig. 2. PN model for a single-arm cluster tool with n steps.

The preset of transition t is the set of all input places to t, i.e., ${}^\bullet t = \{p : p \in P \text{ and } I(p, t) > 0\}$. Its postset is the set of all output places from t, i.e., $t^\bullet = \{p : p \in P \text{ and } O(p, t) > 0\}$. Similarly, p's preset ${}^\bullet p = \{t \in T : O(p, t) > 0\}$ and postset $p^\bullet = \{t \in T : I(p, t) > 0\}$. The transition enabling and firing rules can be found in [19] and [41].

The wafer flow pattern can be denoted as $(m_1, m_2, \ldots, m_n)$ [1], [55], where n is the number of steps for processing a wafer and m_i is the number of PMs used to process wafers at Step i, $i \in \mathbf{N}_n = \{1, 2, \ldots, n\}$. Let $\boldsymbol{\Omega} = \{0\} \cup \mathbf{N}_n$. Based on the wafer flow pattern, the PN model [32], [33] for a single-arm cluster tool is shown in Fig. 2 with the meaning of places and transitions being presented in Table I.

By $K(p_0) = m_0 = \infty$, we mean that the loadlocks can handle all raw and finished wafers in a tool. With a backward strategy, m_i wafers are being processed at Step i, $i \in \mathbf{N}_{\boldsymbol{n}}$. Thus, without loss of generality, we let $M_0(p_i) = m_i$, $i \in \mathbf{N}_{\boldsymbol{n}}$, $M_0(r) = 1$ to indicate that the robot is idle, and $M_0(p_0) = n$ to indicate that there are always wafers to be processed. To avoid deadlock [56]–[58], we give a control policy to make the PN model live [33].

Definition 1 [33]: At marking M, transition y_{i1}, $i \in \mathbf{N}_{\boldsymbol{n-1}} \cup \{0\}$ is said to be control-enabled if $M(p_{i+1}) = m_{i+1} - 1$; and y_{n1} is said to be control-enabled if $M(p_i) = m_i$, $i \in \mathbf{N}_{\boldsymbol{n}}$.

Under the control policy given in Definition 1, the PN is shown to be deadlock-free [33].

B. Modeling Activity Time

In the developed PN model, time is associated with both places and transitions. Time duration $[\zeta_1, \zeta_2]$ is used to denote a robot task's time interval. The wafer processing time is denoted as $([\zeta_1, \zeta_2], \delta)$ which indicates that after the completion of a wafer with $\zeta \in [\zeta_1, \zeta_2]$ time units at Step i, the longest time delay in its corresponding PM must be no more than δ. For a robot task or wafer processing at a PM, $\zeta \in [\zeta_1, \zeta_2]$ is obtained by measuring the real-time

TABLE I
MEANING OF PLACES AND TRANSITIONS OF PN IN FIG. 2

Transition or place	Meaning
$p_0 \in P$	The loadlocks called Step 0 with $K(p_0) = m_0$
$p_i \in P$	The PMs for Step i with $K(p_i) = m_i$, $i \in \mathbf{N}_n$
$r \in P$	The single-arm robot with $K(r) = 1$
$q_{i1} \in P$	The scheduled robot waiting before loading a wafer to Step i, $i \in \mathbf{\Omega}$
$q_{i2} \in P$	The scheduled robot waiting before unloading a wafer from Step i, $i \in \mathbf{\Omega}$
$q_{i3} \in P$	The unscheduled robot waiting before unloading a wafer from Step i, $i \in \mathbf{\Omega}$
$s_{i1} \in T$	Loading a wafer into a PM at Step i modeled by p_i, $i \in \mathbf{N}_n$
$s_{01} \in T$	Loading a completed wafer into a loadlock modeled by p_0
$s_{i2} \in T$	Unloading a wafer from a PM at p_i and moving to p_{i+1}, $i \in \mathbf{N}_{n-1}$
$s_{02} \in T$	Unloading a wafer from a loadlock at p_0 and moving to p_1
$s_{n2} \in T$	Unloading a wafer from p_n and moving to a loadlock
$y_{i1} \in T$	The robot moves from p_{i+2} to p_i without carrying a wafer, $i \in \mathbf{N}_{n-2} \cup \{0\}$
$y_{(n-1)1} \in T$	The robot moves from p_0 to p_{n-1} without carrying a wafer
$y_{n1} \in T$	The robot moves from p_1 to p_n without carrying a wafer

operational time. If $\zeta \in [\zeta_1, \zeta_2]$ represents a scheduled robot waiting time, ζ can be set to be any number in $[\zeta_1, \zeta_2]$. However, if $\zeta \in [\zeta_1, \zeta_2]$ represents an unscheduled robot waiting time, ζ is obtained by real-time measurement and is variable. The time durations for different transitions and places are shown in Table II. Note that the time taken for the robot to move from one PM to another is same under normal conditions defined later.

With wafer residency time constraints, we need to define the liveness for the PN shown in Fig. 2. Let τ_i denote the sojourn time of a token in p_i and ς_i be a sample in $[a_i, b_i]$. Then, the liveness condition of the PN for single-arm cluster tools with residency time constraints can be defined.

Definition 2 [33]: A PN for single-arm cluster tools with residency time constraints is said to be live, if at any marking reached and for any wafer in p_i, $\forall i \in \mathbf{N}_n$, and any ς_i sampled in $[a_i, b_i]$ such that whenever s_{i2} is enabled, $\tau_i - \varsigma_i \le \delta_i$ holds.

C. RCP

To understand the activity time variation, we may view a system as if it operates under normal conditions with random disturbance. A cluster tool is said to be operated under normal conditions if, for any activity with time duration $[\zeta_1, \zeta_2]$, it takes $\zeta = \zeta_1$ time units only. With this definition, in a real-time, the time needed for an activity can be denoted as $\zeta_1 + \Delta\zeta$ with $\zeta_1 \le \zeta_1 + \Delta\zeta \le \zeta_2$. In this way, nonzero $\Delta\zeta$ can be seen as a disturbance and ζ can be any number in $[\zeta_1, \zeta_2]$. In this way, to obtain a feasible periodic schedule under normal conditions is to determine ω_{i1} and ω_{i2} such that $a_i \le \tau_i \le a_i + \delta_i$, $\forall i \in \mathbf{N}_n$.

Let ζ^j denote the actual time taken by completing the jth activity. The random activity time variation can be seen as random disturbance to normal conditions. Thus, we let $\mu^j_{y_{i1}} = \alpha + \sigma^j_{y_{i1}}$ and $\lambda^j_{i1} = c + \rho^j_{i1}$ for all j and $i \in \mathbf{\Omega}$; $\lambda^j_{02} = c_0 + \alpha + \rho^j_{02}$ for all j; $\lambda^j_{i2} = c + \alpha + \rho^j_{i2}$ for all j and $i \in \mathbf{N}_n$. We then dynamically regulate ω_{i2}'s and ω_{i1}'s so as to adapt to the random disturbance based on the real-time observation. With $\sigma^j_{y_{i1}}$, ρ^j_{i2}, and ρ^j_{i1} observed in real-time, if there exists a nonzero value of $\sigma^j_{y_{i1}}$, ρ^j_{i2}, and ρ^j_{i1}, the robot waiting time in q_{i2} and q_{i1} can be shortened by adjusting ω_{i2} and ω_{i1} on-line. We have the following RCP [33].

1) Under the normal conditions, find a periodic schedule by determining ω_{i2} and ω_{i1}, $i \in \mathbf{\Omega}$.
2) Transition s_{01} is fired if the jth token stays in q_{01} for $\omega^j_{01} = \max\{(\omega_{01} - \rho^j_{n2}), 0\}$ time units, and transition s_{i1} is fired if the jth token stays in q_{i1} for $\omega^j_{i1} = \max\{(\omega_{i1} - \rho^j_{(i-1)2}), 0\}$, $i \in \mathbf{N}_n$.
3) Transition y_{i2} is fired if the jth token stays in q_{i2} for $\omega^j_{i2} = \max\{(\omega_{i2} - \sigma^j_{y_{i1}}), 0\}$, $i \in \mathbf{\Omega}$.
4) Transitions s_{i2}, and y_{i1} fire once they are enabled.

By RCP, s_{i2} can fire when there is a token in q_{i3} and a wafer (token) in p_i is completed. This implies that the token waiting time ω^j_{i3} in q_{i3} (or firing s_{i2}) depends on whether a wafer in p_i is completed or not.

III. EXACT UPPER BOUND OF SOJOURN TIME DELAY

A. Effect of Activity Time Variation

Under the normal conditions, a single-arm cluster tool with residency time constraints should be scheduled such that $a_i \le \tau_i \le a_i + \delta_i$. Thus, we need to know how the activity time variation affects the wafer sojourn time delay in a PM. We first summarize the results of wafer sojourn time delay caused by activity time variation as obtained in [32].

Under the normal conditions, ω_{i2} and ω_{i1}, $i \in \mathbf{\Omega}$, are the scheduled waiting time and are constant, while ω_{i3}, $i \in \mathbf{\Omega}$, should be zero. Hence, $\lambda^j_{02} = c_0 + \alpha$ for all j; $\lambda^j_{d2} = c + \alpha$ for all j and $d \in \mathbf{N}_n$; $\mu^j_{y_{d1}} = \alpha$, $\omega^j_{d2} = \omega_{d2}$, $\omega^j_{d1} = \omega_{d1}$ and $\lambda^j_{d1} = c$ for all j and d. Thus, at any steady state marking M,

TABLE II
TIME DURATIONS ASSOCIATED WITH TRANSITIONS AND PLACES

Symbol	Transition or place	Actions	Allowed time duration
λ_{i1}	$s_{i1}\in T$	Robot loads a wafer into Step i, $i\in\mathbf{\Omega}$	$[c, d]$
λ_{i2}	$s_{i2}\in T$	Robot unloads a wafer from Step i and moves to p_{i+1}, $i\in\mathbf{N_{n-1}}$	$[c+\alpha, d+\beta]$
	$s_{n2}\in T$	Robot unloads a wafer from Step n and moves to a loadlock	
λ_{02}	$s_{02}\in T$	Robot unloads a wafer from a loadlock, aligns it, and moves to p_1	$[c_0+\alpha, d_0+\beta]$
$\mu_{y_{i1}}$	$y_{i1}\in T$	Robot moves from Steps $i+2$ to i, $i\in\mathbf{N_{n-2}}\cup\{0\}$	
$\mu_{y_{(n-1)1}}$	$y_{(n-1)1}\in T$	Robot moves from Steps 0 to n-1	$[\alpha, \beta]$
$\mu_{y_{n1}}$	$y_{n1}\in T$	Robot moves from Steps 1 to n	
τ_i	$p_i\in P$	A wafer being processed and waiting in p_i, $i\in\mathbf{N_n}$	$([a_i, b_i], \delta_i)$
ω_{i2}	$q_{i2}\in P$	Scheduled robot waiting before unloading a wafer from Step i, $i\in\mathbf{\Omega}$	
ω_{i1}	$q_{i1}\in P$	Scheduled robot waiting before loading a wafer to Step i, $i\in\mathbf{\Omega}$	$[0, \infty]$
ω_{i3}	$q_{i2}\in P$	Unscheduled robot waiting before unloading a wafer from Step i, $i\in\mathbf{\Omega}$	

we have

$$\begin{aligned}\tau_1 &= m_1\times\left[2(n+1)\alpha+(2n+1)c+c_0+\sum\nolimits_{d=0}^{n}\omega_{d2}+\sum\nolimits_{d=0}^{n}\omega_{d1}\right]\\ &\quad-(3c+c_0+3\alpha+\omega_{02}+\omega_{11}+\omega_{21})\\ &= m_1\times\psi-(3c+c_0+3\alpha+\omega_{02}+\omega_{11}+\omega_{21}) \qquad (1)\\ \tau_i &= m_i\times\left[2(n+1)\alpha+(2n+1)c+c_0+\sum\nolimits_{d=0}^{n}\omega_{d2}+\sum\nolimits_{d=0}^{n}\omega_{d1}\right]\\ &\quad-\left(4c+3\alpha+\omega_{(i-1)2}+\omega_{i1}+\omega_{(i+1)1}\right)\\ &= m_i\times\psi-\left(4c+3\alpha+\omega_{(i-1)2}+\omega_{(i+1)1}+\omega_{i1}\right)\\ &\quad i=2,3,\ldots,n-1 \qquad (2)\\ \tau_n &= m_n\times\left[2(n+1)\alpha+(2n+1)c+c_0+\sum\nolimits_{d=0}^{n}\omega_{d2}+\sum\nolimits_{d=0}^{n}\omega_{d1}\right]\\ &\quad-\left(4c+3\alpha+\omega_{(n-1)2}+\omega_{n1}+\omega_{01}\right)\\ &= m_n\times\psi-\left(4c+3\alpha+\omega_{(n-1)2}+\omega_{n1}+\omega_{01}\right). \qquad (3)\end{aligned}$$

Also, under normal conditions, the robot cycle time is

$$\begin{aligned}\psi &= 2(n+1)\alpha+(2n+1)c+c_0+\sum\nolimits_{d=0}^{n}\omega_{d2}+\sum\nolimits_{d=0}^{n}\omega_{d1}\\ &= \psi_1+\psi_2 \qquad (4)\end{aligned}$$

where $\psi_1 = 2(n+1)\alpha+(2n+1)c+c_0$ is a constant and known in advance and $\psi_2=\sum_{d=0}^{n}\omega_{d2}+\sum_{d=0}^{n}\omega_{d1}$ is to be determined by a schedule. It should be noticed that ψ is independent of the wafer processing time.

Let $\theta_1=(\tau_1+3c+c_0+3\alpha+\omega_{02}+\omega_{11}+\omega_{21})/(m_1)$, $\theta_i=(\tau_i+4c+3\alpha+\omega_{(i-1)2}+\omega_{i1}+\omega_{(i+1)1})/(m_i)$, $i\in\mathbf{N_{n-1}}\backslash\{1\}$, and $\theta_n=(\tau_n+4c+3\alpha+\omega_{(n-1)2}+\omega_{n1}+\omega_{01})/(m_n)$ denote the cycle time for Step i, $i\in\mathbf{N_n}$. Further, let θ be the production cycle time of the system. Since the process of single-arm cluster tools is a serial one, the production rate is same for all the steps and this production rate is the cycle time for the system. We have the following proposition.

Proposition 1: In the steady state, a single-arm cluster tool with a backward strategy has the same cycle time for all processing steps, that is

$$\theta=\theta_1=\theta_2=\cdots=\theta_n. \qquad (5)$$

Then, the relationship between the production cycle and robot cycle can be analyzed based on the model shown in Fig. 2. Assume that wafer W_k is loaded into Step i at time τ_k and W_{k+1} is loaded into it at τ_{k+1}. Then, $[\tau_k, \tau_{k+1}]$ forms a cycle for Step i. During this time, s_{i1} fires twice, and the robot completes the following activities: firing $s_{i1}\rightarrow y_{(i-2)1}\rightarrow$ waiting in $q_{(i-2)2}\rightarrow s_{(i-2)2}\rightarrow$ waiting in $q_{(i-1)1}\rightarrow s_{(i-1)1}\rightarrow y_{(i-3)1}\rightarrow\ldots\rightarrow y_{01}\rightarrow$ waiting in $q_{02}\rightarrow s_{02}\rightarrow$ waiting in $q_{11}\rightarrow s_{11}\rightarrow y_{n1}\rightarrow$ waiting in $q_{n2}\rightarrow s_{n2}\rightarrow$ waiting in $q_{01}\rightarrow s_{01}\rightarrow y_{(n-1)1}\rightarrow\ldots\rightarrow y_{i1}\rightarrow$ waiting in $q_{i2}\rightarrow s_{i2}\rightarrow$ waiting in $q_{(i+1)1}\rightarrow s_{(i+1)1}\rightarrow y_{(i-1)1}\rightarrow$ waiting in $q_{(i-1)2}\rightarrow s_{(i-1)2}\rightarrow$ waiting in $q_{i1}\rightarrow s_{i1}$ again. Note that, during this time, the robot completes exactly one cycle. Thus, we have the following proposition.

Proposition 2: In the steady state, under the normal conditions, a single-arm cluster tool with a backward strategy has the same cycle time for the robot and each step, that is

$$\theta=\theta_1=\theta_2=\cdots=\theta_n=\psi. \qquad (6)$$

According to (4), α, c, and c_0 are all deterministic, while ω_{d2} and ω_{d1}, $d\in\mathbf{\Omega}$, are changeable, i.e., ψ_1 is deterministic while the robot waiting time in ψ_2 can be regulated. Thus, to schedule the system under the normal conditions is to appropriately set ω_{d2} and ω_{d1}, $d\in\mathbf{\Omega}$, such that (6) holds and at the same time the wafer residency time constraints are satisfied. With activity time variation considered, there may exist a nonzero value of $\sigma_{y_{d1}}^{j}$, ρ_{d2}^{j}, and ρ_{d1}^{j} obtained by real-time measurement. Thus, RCP reduces the effects of the time variation on the wafer sojourn time delay as much as possible. Let η_{d2}^{j} be the time delay that is caused by $\sigma_{y_{d1}}^{j}$ and can be offset by adjusting ω_{d2}, and η_{01}^j and η_{i1}^j be the time delay that is caused by ρ_{n2}^j and $\rho_{(i-1)2}^j$ and can be offset by adjusting ω_{01} and ω_{i1}, respectively. Then, we have $\omega_{i2}^j + \sigma_{y_{i1}}^j = \omega_{i2} + \eta_{i2}^j$, or $\eta_{i2}^j = \max\{(\sigma_{y_{i1}}^j - \omega_{i2}), 0\}$. In this way, the effect of $\sigma_{y_{d1}}^j$ on τ_i can be made as small as possible. Similarly, we have $\omega_{01}^j + \rho_{n2}^j = \omega_{01} + \eta_{01}^j$ and $\omega_{i1}^j + \rho_{(i-1)2}^j = \omega_{i1} + \eta_{i1}^j$, or $\eta_{01}^j = \max\{(\rho_{n2}^j - \omega_{01}), 0\}$ and $\eta_{i1}^j = \max\{(\rho_{(i-1)2}^j - \omega_{i1}), 0\}$. Then, according to [32], the wafer sojourn time in p_i is

$$\begin{aligned}\tau_1 &= m_1 \times \left[2(n+1)\alpha + (2n+1)c + c_0 + \sum_{d=0}^{n}\omega_{d2} + \sum_{d=0}^{n}\omega_{d1}\right]\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_1-1}\eta_{d2}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_1-1}\eta_{d1}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_1-1}\rho_{d1}^j\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_1-1}\omega_{d3}^j - \Big(3c + c_0 + 3\alpha + \omega_{02} + \omega_{11} + \omega_{21}\\ &\qquad + \omega_{02}^k + \omega_{11}^k + \omega_{21}^k - \omega_{02} - \omega_{11}\\ &\qquad - \omega_{21} + \omega_{03}^k + \rho_{11}^k + \rho_{12}^k\\ &\qquad + \rho_{21}^k + \rho_{02}^k + \sigma_{y_{01}}^k\Big)\\ &= \Lambda_1 + \Theta_1 \end{aligned} \tag{7}$$

$$\begin{aligned}\tau_i &= m_i \times \left[2(n+1)\alpha + (2n+1)c + c_0 + \sum_{d=0}^{n}\omega_{d2} + \sum_{d=0}^{n}\omega_{d1}\right]\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d2}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d1}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\rho_{d1}^j\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega_{d3}^j - \Big(4c + 3\alpha + \omega_{(i-1)2} + \omega_{i1} + \omega_{(i+1)1}\\ &\qquad + \omega_{(i-1)2}^k + \omega_{i1}^k + \omega_{(i+1)1}^k - \omega_{(i-1)2}\\ &\qquad - \omega_{i1} - \omega_{(i+1)1} + \omega_{(i-1)3}^k + \rho_{i1}^k\\ &\qquad + \rho_{i2}^k + \rho_{(i+1)1}^k + \rho_{(i-1)2}^k + \sigma_{y_{(i-1)1}}^k\Big)\\ &= \Lambda_i + \Theta_i,\ 1 < i < n \end{aligned} \tag{8}$$

$$\begin{aligned}\tau_n &= m_n \times \left[2(n+1)\alpha + (2n+1)c + c_0 + \sum_{d=0}^{n}\omega_{d2} + \sum_{d=0}^{n}\omega_{d1}\right]\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_n-1}\eta_{d2}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_n-1}\eta_{d1}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_n-1}\rho_{d1}^j\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_n-1}\omega_{d3}^j - \Big(4c + 3\alpha + \omega_{(n-1)2} + \omega_{n1} + \omega_{01}\\ &\qquad + \omega_{(n-1)2}^k + \omega_{n1}^k + \omega_{01}^k - \omega_{(n-1)2}\\ &\qquad - \omega_{n1} - \omega_{01} + \omega_{(n-1)3}^k + \rho_{n1}^k\\ &\qquad + \rho_{n2}^k + \rho_{01}^k + \rho_{(n-1)2}^k + \sigma_{y_{(n-1)1}}^k\Big)\\ &= \Lambda_n + \Theta_n \end{aligned} \tag{9}$$

where $\Lambda_1 = m_1 \times \psi - (3c + c_0 + 3\alpha + \omega_{02} + \omega_{11} + \omega_{21})$, $\Lambda_i = m_i \times \psi - (4c + 3\alpha + \omega_{(i-1)2} + \omega_{i1} + \omega_{(i+1)1})$, $i \in \mathbf{N_{n-1}}\backslash\{1\}$, and $\Lambda_n = m_n \times \psi - (4c + 3\alpha + \omega_{(n-1)2} + \omega_{n1} + \omega_{01})$ are the scheduled sojourn time under the normal conditions given by (1)–(3) and are constant when the periodic schedule is determined.

Now, the sojourn time disturbance is given as

$$\begin{aligned}\Theta_1 &= \sum_{d=0}^{n}\sum_{j=k}^{k+m_1-1}\eta_{d2}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_1-1}\eta_{d1}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_1-1}\rho_{d1}^j\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_1-1}\omega_{d3}^j - \Big(\omega_{02}^k + \omega_{11}^k + \omega_{21}^k - \omega_{02} - \omega_{11}\\ &\qquad - \omega_{21} + \omega_{03}^k + \rho_{11}^k + \rho_{12}^k\\ &\qquad + \rho_{21}^k + \rho_{02}^k + \sigma_{y_{01}}^k\Big)\end{aligned}$$

$$\begin{aligned}\Theta_i &= \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d2}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\eta_{d1}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\rho_{d1}^j\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_i-1}\omega_{d3}^j - \Big(\omega_{(i-1)2}^k + \omega_{i1}^k + \omega_{(i+1)1}^k - \omega_{(i-1)2}\\ &\qquad - \omega_{i1} - \omega_{(i+1)1} + \omega_{(i-1)3}^k + \rho_{i1}^k\\ &\qquad + \rho_{i2}^k + \rho_{(i+1)1}^k + \rho_{(i-1)2}^k\\ &\qquad + \sigma_{y_{(i-1)1}}^k\Big),\ i \in \mathbf{N_{n-1}}\backslash\{1\}\end{aligned}$$

$$\begin{aligned}\Theta_n &= \sum_{d=0}^{n}\sum_{j=k}^{k+m_n-1}\eta_{d2}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_n-1}\eta_{d1}^j + \sum_{d=0}^{n}\sum_{j=k}^{k+m_n-1}\rho_{d1}^j\\ &\quad + \sum_{d=0}^{n}\sum_{j=k}^{k+m_n-1}\omega_{d3}^j - \Big(\omega_{(n-1)2}^k + \omega_{n1}^k + \omega_{01}^k - \omega_{(n-1)2}\\ &\qquad - \omega_{n1} - \omega_{01} + \omega_{(n-1)3}^k + \rho_{n1}^k\\ &\qquad + \rho_{n2}^k + \rho_{01}^k + \rho_{(n-1)2}^k + \sigma_{y_{(n-1)1}}^k\Big).\end{aligned}$$

Note that η_{d2}^j, η_{d1}^j, ρ_{i1}^k, ρ_{i2}^k, $\rho_{(i+1)1}^k$, $\rho_{(i-1)2}^k$, and $\sigma_{y_{i-1}}^k$ are obtained via real-time observation, while ω_{d1} and ω_{d2} are determined by an off-line schedule and thus known in advance. ω_{d3}^j is uncontrollable and varies with j. In fact, Θ_i represents the accumulated robot time delay when the robot arrives at q_{i3} for unloading a completed wafer there. Under the normal conditions, the necessary and sufficient schedulability conditions are presented in [3]. Thus, if we can find a method to obtain Θ_i for the worst case, with the results in [3], can we find the necessary and sufficient schedulability conditions for single-arm cluster tools with wafer residency time constraints and activity time variation? To answer it, we are required to find the exact upper bound of the wafer sojourn time delay.

B. Computing Exact Upper Bound

With wafer flow pattern $(m_1, m_2, \ldots, m_n)$, $m = m_1 + m_2 + \cdots + m_n$ wafers are being processed concurrently. We number them as W_1–W_m. Let Ξ_i denote the set of wafers that are being processed in p_i. Further, let $E1 = m_2 + \cdots + m_n + 1$, $L1 = m$, $Ei = m_{i+1} + \cdots + m_n + 1$, $Li = m_i + \cdots + m_n$, $i \in \mathbf{N_{n-1}}$, $En = 1$, and $Ln = m_n$, such that W_{Ei} and W_{Li} are the earliest and latest wafers released into p_i, respectively. Let $Ei\_j = Ei + j$, then, we have $\Xi_i = \{W_{Ei}, W_{Ei\_1}, \ldots, W_{Li}\}$, $i \in \mathbf{N}_n$. Assume that it takes $\upsilon_i \in [a_i, b_i]$ time units to complete W_{Ei}, leading to a time delay $\max\{(\upsilon_i - \Lambda_i), 0\}$. Let $H_i = \max\{(b_i - \Lambda_i), 0\}$, $i \in \mathbf{N}_n$, be the longest time delay caused by processing a wafer at p_i and $H_0 = 0$ since there is no processing time delay at Step 0. Further let $\eta_{11} = \max\{(d_0 - c_0) + (\beta - \alpha) - \omega_{11}, 0\}$, $\eta_{i1} = \max\{(d - c) + (\beta - \alpha) - \omega_{i1}, 0\}$, and $\eta_{i2} = \max\{(\beta - \alpha) - \omega_{i2}, 0\}$.

The infeasibility of a schedule is caused by delay τ_i, $i \in \mathbf{N}_n$. It follows from (7)–(9) that $\tau_i = \Lambda_i + \Theta_i$, where Λ_i is the robot task time in a cycle under the normal conditions, while Θ_i is the accumulated robot time delay. Under the normal conditions, the system can be scheduled such that $a_i \le \tau_i = \Lambda_i \le a_i + \delta_i$. With activity time variation, it is required that $\tau_i = \Lambda_i + \Theta_i \le a_i + \Delta a_i + \delta_i$, where $\Delta a_i \in [0, b_i - a_i]$. To do so, with Λ_i being known, we have to find Θ_i. When Θ_i reaches its largest value, or the upper bound, in the worst case, if $\tau_i = \Lambda_i + \Theta_i \le a_i + \Delta a_i + \delta_i$ holds, the system operates in a feasible state. Thus, the key is to calculate the upper bound of Θ_i. Notice that the worst case occurs when Δa_i is zero such that the robot does not need to wait at q_{i3} for unloading a processed wafer from p_i since $a_i \le \Lambda_i$, or $\omega_{d3}^{m_i} = 0$. Thus, to check the feasibility of a schedule, we need to find the exact upper bound of $B_i = \Theta_i - \omega_{d3}^{m_i}$. To do so, based on the PN model, we analyze the fabrication process as follows.

After loading a wafer W_{Ei} into p_i, the robot goes to Step $i-2$ by firing $y_{(i-2)1}$ for unloading a processed wafer there. Then, a sequence of tasks is executed. Finally, the robot comes back to Step i for unloading W_{Ei}. This process undergoes m_i robot cycles. This implies that, before the robot comes back to Step i again, it goes through every Step $d \notin \{i, i-1\}$ for m_i times. Hence, to calculate the exact time delay during this process, the robot task sequence can be divided into a number of small segments such that the time delay of each segment can be calculated straightforwardly. Then, the time delay can be calculated in a sequential way. With this idea, we analyze how it can be divided into small segments next.

Let κ_{id}^j denote the transition firing sequence for the robot to go from Steps i to d in the jth robot cycle. After loading a wafer into p_i, starting from Step i, in the first robot cycle, the robot goes to Step $i-2$ by firing $y_{(i-2)1}$. Then, through a number of steps, it goes to Step d for any given $d \in \mathbf{N}_n$ and waits there for unloading a wafer, or a token goes into q_{d3} by executing the following transition sequence:

$\kappa_{id}^1 =$ $\langle$firing $y_{(i-2)1} \to$ waiting in $q_{(i-2)2} \to y_{(i-2)2} \to$ waiting in $q_{(i-2)3} \to s_{(i-2)2} \to$ waiting in $q_{(i-1)1} \to s_{(i-1)1} \to \ldots \to y_{d1} \to$ waiting in $q_{d2} \to y_{d2} \to$ waiting in $q_{d3}\rangle$, $d \in \mathbf{N}_{i-2} \cup \{0\}$

$\kappa_{id}^1 =$ $\langle\kappa_{i0}^1 \to s_{02} \to$ waiting in $q_{11} \to s_{11} \to y_{n1} \to$ waiting in $q_{n2} \to y_{n2} \to$ waiting in $q_{n3}\rangle$, $d = n$, or

$\kappa_{id}^1 =$ $\langle\kappa_{in}^1 \to s_{n2} \to$ waiting in $q_{01} \to s_{01} \to \ldots \to y_{d1} \to$ waiting in $q_{d2} \to y_{d2} \to$ waiting in $q_{d3}\rangle$, $d \in \mathbf{N}_{n-1} \backslash \mathbf{N}_{i-2}$.

Similarly, in the second robot cycle, starting from Step i, the robot goes to Step $i-2$, or a token goes into $q_{(i-2)3}$, by executing the following sequence in the PN in Fig. 2:

$\kappa_{i(i-2)}^2 = \langle\kappa_{i(i-1)}^1 \to s_{(i-1)2} \to$ waiting in $q_{i1} \to s_{i1} \to y_{(i-2)1} \to$ waiting in $q_{(i-2)2} \to y_{(i-2)2} \to$ waiting in $q_{(i-2)3}\rangle$.

After undergoing $(j-1) < m_i$ robot cycles, the robot continues its jth cycle. With $j > 1$, the robot goes to Step d in the jth robot cycle by executing the following sequence:

$\kappa_{id}^j =$ $\langle\kappa_{i(i-1)}^{j-1} \to s_{(i-1)2} \to$ waiting in $q_{i1} \to s_{i1} \to y_{(i-2)1} \to$ waiting in $q_{(i-2)2} \to y_{(i-2)2} \to$ waiting in $q_{(i-2)3} \to s_{(i-2)2} \to$ waiting in $q_{(i-1)1} \to s_{(i-1)1} \to \ldots \to y_{d1} \to$ waiting in $q_{d2} \to y_{d2} \to$ waiting in $q_{d3}\rangle$, $d \in \mathbf{N}_{i-2} \cup \{0\}$ and $1 < j < m_i$

$\kappa_{id}^j =$ $\langle\kappa_{i0}^j \to s_{02} \to$ waiting in $q_{11} \to s_{11} \to y_{n1} \to$ waiting in $q_{n2} \to y_{n2} \to$ waiting in $q_{n3}\rangle$, $d = n$ and $1 < j < m_i$, or

$\kappa_{id}^j =$ $\langle\kappa_{in}^j \to s_{n2} \to$ waiting in $q_{01} \to s_{01} \to \ldots \to y_{d1} \to$ waiting in $q_{d2} \to y_{d2} \to$ waiting in $q_{d3}\rangle$, $d \in \mathbf{N}_{n-1} \backslash \mathbf{N}_{i-2}$ and $1 < j < m_i$.

Similarly,

$\kappa_{id}^{m_i} =$ $\langle\kappa_{i(i-1)}^{m_i-1} \to s_{(i-1)2} \to$ waiting in $q_{i1} \to s_{i1} \to y_{(i-2)1} \to$ waiting in $q_{(i-2)2} \to y_{(i-2)2} \to$ waiting in $q_{(i-2)3} \to s_{(i-2)2} \to$ waiting in $q_{(i-1)1} \to s_{(i-1)1} \to \ldots \to y_{d1} \to$ waiting in $q_{d2} \to y_{d2} \to$ waiting in $q_{d3}\rangle$, $d \in \mathbf{N}_{i-2} \cup \{0\}$,

$\kappa_{id}^{m_i} =$ $\langle\kappa_{i0}^{m_i} \to s_{02} \to$ waiting in $q_{11} \to s_{11} \to y_{n1} \to$ waiting in $q_{n2} \to y_{n2} \to$ waiting in $q_{n3}\rangle$, $d = n$, or

$\kappa_{id}^{m_i} =$ $\langle\kappa_{in}^{m_i} \to s_{n2} \to$ waiting in $q_{01} \to s_{01} \to \ldots \to y_{d1} \to$ waiting in $q_{d2} \to y_{d2} \to$ waiting in $q_{d3}\rangle$, $d \in \mathbf{N}_{n-1} \backslash \mathbf{N}_i$.

Then, after performing $\kappa_{i(i+1)}^{m_i}$, the robot performs task sequence $\kappa_1 = \langle s_{(i+1)2} \to$ waiting in $q_{(i+2)1} \to s_{(i+2)1} \to y_{i1} \to$ waiting in $q_{i2}\rangle$. Let $\|\bullet\|$ denote the exact upper bound of time delay during the execution of an activity sequence and $\Gamma_{id}^j = \|\kappa_{id}^j\|$. Note that the exact upper bound of time delay for executing $\langle\kappa_{i(i+1)}^{m_i} \to \kappa_1\rangle$ is B_i. Because $\|\kappa_1\| = \eta_{(i+2)1} + \rho + \eta_{i2}$ and $\|\kappa_{i(i+1)}^{m_i}\| = \Gamma_{i(i+1)}^{m_i}$, we have $B_i = \Gamma_{i(i+1)}^{m_i} + \eta_{(i+2)1} + \rho + \eta_{i2}$, $i \in \mathbf{N}_{n-2}$. Similarly, we have

$$B_i = \begin{cases} \Gamma_{i(i+1)}^{m_i} + \eta_{(i+2)1} + \rho + \eta_{i2}, i \in \mathbf{N}_{n-2} \\ \Gamma_{in}^{m_{n-1}} + \eta_{01} + \rho + \eta_{(n-1)2}, i = n-1 \\ \Gamma_{i0}^{m_n} + \eta_{11} + \rho + \eta_{n2}, i = n. \end{cases} \quad (10)$$

The remaining problem is how to calculate $\Gamma_{i(i+1)}^{m_i}$, $i \in \mathbf{N}_n \backslash \{n\}$, or $\Gamma_{i0}^{m_n}$, $i = n$. To do so, we divide it into several cases. First, we consider the case when $m_i \le m_u$ for any $i \ne u$. Without loss of generality, for Step i, we analyze the longest sojourn time delay of W_{Ei}. We have $\kappa_{i(i+1)}^{m_i} = \langle\kappa_{i(i+2)}^{m_i} \to \kappa_2 \to$ waiting in $q_{(i+1)3}\rangle$ with $\kappa_2 = \langle s_{(i+2)2} \to$ waiting in $q_{(i+3)1} \to s_{(i+3)1} \to y_{(i+1)1} \to$ waiting in $q_{(i+1)2} \to$ firing $y_{(i+1)2}\rangle$, or $\kappa_{i(i+1)}^{m_i}$ can be divided into $\kappa_{i(i+2)}^{m_i}$, κ_2, and $\langle$waiting in $q_{(i+1)3}\rangle$. By the RCP and that firing $y_{(i+1)2}$ takes no time, we have $\|\kappa_2\| = \eta_{(i+3)1} + \rho + \eta_{(i+1)2}$. Thus, $\|\kappa_{i(i+1)}^{m_i}\| = \|\kappa_{i(i+2)}^{m_i} \to \kappa_2\| = \Gamma_{i(i+2)}^{m_i} + \eta_{(i+3)1} + \rho + \eta_{(i+1)2}$. Note that, under the normal conditions, the robot can unload the wafer from PM_{i+1} immediately after performing κ_2. However, with activity time variation, it should go to $q_{(i+1)3}$ for waiting after performing κ_2, since there may be a disturbance on wafer processing in PM_{i+1}. W_{Ei} and $W_{E(i+1)}$ are the first wafers loaded into Steps i and $i+1$ in different cycles, respectively, if $m_i \neq m_{(i+1)}$. When $m_i = m_{i+1}$, W_{Ei} and $W_{E(i+1)}$ are loaded into Steps i and $i+1$ in the same cycle. Then, with a backward strategy, it follows from $m_i \leq m_u$ for any $i \neq u$ that wafer $W_{E(i+1)}$ is loaded into Step $i+1$ before W_{Ei} into Step i.

With the above analysis, in order to analyze the exact longest activity time delay on wafer sojourn time at Step i, we should calculate the robot's accumulated activity time delay by starting from the time when W_{Ei} has just been loaded into Step i. Thus, when the robot goes to $q_{(i+1)3}$ in the m_ith cycle, we should check if the robot waiting in $q_{(i+1)3}$ is necessary. In other words, at this time, the robot should wait in $q_{(i+1)3}$ for $\max\{H_{i+1} - (\Gamma^{m_i}_{i(i+2)} + \eta_{(i+3)1} + \rho + \eta_{(i+1)2}), 0\}$ time units. After the robot leaves $q_{(i+1)3}$, its accumulated activity time delay is $\Gamma^{m_i}_{i(i+1)} = \max\{\Gamma^{m_i}_{i(i+2)} + \eta_{(i+3)1} + \rho + \eta_{(i+1)2}, H_{i+1}\}$. For the sake of clarity, it is assumed that $m_n \leq m_u$, $u \in \mathbf{N_{n-1}}$, holds. Then, for Step n, we analyze the longest sojourn time delay of W_{En}. The loadlocks, or Step 0 can hold all the wafers, and this is equivalent to $m_0 \geq \max\{m_j, j \in \mathbf{N_n}\}$. In order to calculate B_n, the key is to calculate $\Gamma^{m_n}_{n0}$. Thus, we have $\kappa^{m_n}_{n0} = \langle \kappa^{m_n}_{n1} \rightarrow \kappa_2 \rightarrow$ waiting in $q_{03}\rangle$ with $\kappa_2 = \langle s_{12} \rightarrow$ waiting in $q_{21} \rightarrow s_{21} \rightarrow y_{01} \rightarrow$ waiting in $q_{02} \rightarrow$ firing $y_{02}\rangle$, or $\kappa^{m_n}_{n0}$ can be divided into $\kappa^{m_n}_{n1}$, κ_2, and $\langle$waiting in $q_{03}\rangle$.

By RCP, firing y_{02} takes no time. Thus we have $\|\kappa_2\| = \eta_{21} + \rho + \eta_{02}$. Hence, $\|\kappa^{m_n}_{n1} \rightarrow \kappa_2\| = \Gamma^{m_n}_{n1} + \eta_{21} + \rho + \eta_{02}$. It is known that, under the normal conditions, the robot can unload a wafer from Step 0 (loadlocks) immediately after performing κ_2. Then, with activity time variation, we have to check if the robot task delay caused by $\kappa^{m_n}_{n1} \rightarrow \kappa_2$ is larger than the processing time delay at Step 0. After the robot leaves q_{03}, its accumulated activity time delay is $\Gamma^{m_n}_{n0} = \max\{\Gamma^{m_n}_{n1} + \eta_{21} + \rho + \eta_{02}, H_0\}$. In fact, $H_0 = 0$ leads to $\Gamma^{m_n}_{n0} = \Gamma^{m_n}_{n1} + \eta_{21} + \rho + \eta_{02}$. This implies that we can obtain $\Gamma^{m_n}_{n0}$ by calculating $\Gamma^{m_n}_{n1}$ first. Similarly, $\Gamma^{m_n}_{n1} = \max\{\Gamma^{m_n}_{n2} + \eta_{31} + \rho + \eta_{12}, H_1\}$ implies that we can obtain $\Gamma^{m_n}_{n1}$ by calculating $\Gamma^{m_n}_{n2}$ first, $\ldots$, $\Gamma^{m_n}_{nk} = \max\{\Gamma^{m_n}_{n(k+1)} + \eta_{(k+2)1} + \rho + \eta_{k2}, H_k\}$, $0 \leq k \leq n-3$, implies that we can obtain $\Gamma^{m_n}_{nk}$ by calculating $\Gamma^{m_n}_{n(k+1)}$ first.

Note that $\kappa^{m_n}_{n(n-2)}$ can be divided into $\kappa^{m_n-1}_{n(n-1)}$, $\kappa_3 = \langle s_{(i-1)2} \rightarrow$ waiting in $q_{i1} \rightarrow s_{i1} \rightarrow y_{(i-2)1} \rightarrow$ waiting in $q_{(i-2)2} \rightarrow$ firing $y_{(i-2)2}\rangle$, and $\langle$waiting in $q_{(i-2)3}\rangle$, since the robot performs $\langle\kappa^{m_n-1}_{n(n-1)} \rightarrow \kappa_3 \rightarrow$ waiting in $q_{(i-2)3}\rangle$ in the (m_n-1)th cycle. With RCP, $\|\kappa^{m_n-1}_{n(n-1)} \rightarrow \kappa_3\| = \Gamma^{m_n-1}_{n(n-1)} + \eta_{n1} + \rho + \eta_{(n-2)2}$. Thus, $\Gamma^{m_n}_{n(n-2)} = \max\{\Gamma^{m_n-1}_{n(n-1)} + \eta_{n1} + \rho + \eta_{(n-2)2}, H_{(n-2)}\}$. Then, $\Gamma^{m_n-1}_{n(n-1)} = \max\{\Gamma^{m_n-1}_{nn} + \eta_{01} + \rho + \eta_{(n-1)2}, H_{(n-1)}\}$ and $\Gamma^{m_n-1}_{nn} = \max\{\Gamma^{m_n-1}_{n0} + \eta_{11} + \rho + \eta_{n2}, H_n\}$. Furthermore, we have $\Gamma^{j}_{nk} = \max\{\Gamma^{j}_{n(k+1)} + \eta_{(k+2)1} + \rho + \eta_{k2}, H_k\}$, $0 \leq k \leq n-3$ and $2 \leq j \leq m_n - 1$, and $\Gamma^{j}_{n(n-2)} = \max\{\Gamma^{j-1}_{n(n-1)} + \eta_{n1} + \rho + \eta_{(n-2)2}, H_{(n-2)}\}$. By continuously doing so, we have $\Gamma^{2}_{n(n-2)} = \max\{\Gamma^{1}_{n(n-1)} + \eta_{n1} + \rho + \eta_{(n-2)2}, H_{(n-2)}\}$, $\Gamma^{1}_{n(n-1)} = \max\{\Gamma^{1}_{nn} + \eta_{01} + \rho + \eta_{(n-1)2}, H_{(n-1)}\}$, and $\Gamma^{1}_{nn} = \max\{\Gamma^{1}_{n0} + \eta_{11} + \rho + \eta_{n2}, H_n\}$. Generally, we have $\Gamma^{1}_{nk} = \max\{\Gamma^{1}_{n(k+1)} + \eta_{(k+2)1} + \rho + \eta_{k2}, H_k\}$, $0 \leq k \leq n-4$, and $\Gamma^{1}_{n(n-3)} = \max\{\Gamma^{1}_{n(n-2)} + \eta_{(n-1)1} + \rho + \eta_{(n-3)2}, H_{(n-3)}\}$. According to [23], the activity time variation before loading wafer W_{En} has no effect on the wafer sojourn time of W_{En} at Step n. Thus, we have $\Gamma^{1}_{n(n-2)} = \max\{\eta_{(n-2)2}, H_{n-2}\}$ where $\eta_{(n-2)2}$ and H_{n-2} are known in advance. This implies that $\Gamma^{m_n}_{n0}$ can be computed by calculating $\Gamma^{1}_{n(n-2)}$, $\Gamma^{1}_{n(n-3)}$, $\ldots$, Γ^{1}_{n0}, Γ^{1}_{nn}, $\Gamma^{1}_{n(n-1)}$, $\Gamma^{2}_{n(n-2)}$, $\ldots$, $\Gamma^{j}_{n(n-2)}$, $\Gamma^{j}_{n(n-3)}$, $\ldots$, Γ^{j}_{n0}, Γ^{j}_{nn}, $\Gamma^{j}_{n(n-1)}$, $\Gamma^{j+1}_{n(n-2)}$, $\ldots$, $\Gamma^{m_n}_{n(n-2)}$, $\Gamma^{m_n}_{n(n-3)}$, $\ldots$, and $\Gamma^{m_n}_{n0}$ in a sequential way. Then, by (10), we can obtain B_n. Note that, in a cluster tool, there are at least two steps, or $n \geq 2$. Algorithm 1 finds $\Gamma^{m_n}_{n0}$ if $m_n \leq m_u$, $u \in \mathbf{N_{n-1}}$.

Algorithm 1: Calculate $\Gamma^{m_n}_{n0}$ when $m_n \leq m_u$, $u \in \mathbf{N_{n-1}}$

If $m_n \leq m_u$, $u \in \mathbf{N_{n-1}}$, find $\Gamma^{m_n}_{n0}$ as follows.

1) $\Gamma^{1}_{n(n-2)} = \max\{\eta_{(n-2)2}, H_{n-2}\}$;
2) If $n > 2$
3) $\quad k = n - 3$;
4) $\quad$ Otherwise $k = n$;
5) $j = 1$;
6) While $j \leq m_n$
7) $\quad$ While $k \neq n - 2$
8) $\quad\quad$ If $0 \leq k \leq n - 3$
9) $\quad\quad\quad \Gamma^{j}_{nk} = \max\{(\Gamma^{j}_{n(k+1)} + \eta_{(k+2)1} + \rho + \eta_{k2}), H_k\}$;
10) $\quad\quad\quad$ If $j = m_n$ and $k = 0$
11) $\quad\quad\quad\quad$ Go to (24);
12) $\quad\quad$ If $k = n - 1$
13) $\Gamma^{j}_{n(n-1)} = \max\{(\Gamma^{j}_{nn} + \eta_{01} + \rho + \eta_{(n-1)2}), H_{n-1}\}$;
14) $\quad\quad$ If $k = n$
15) $\Gamma^{j}_{nn} = \max\{(\Gamma^{j}_{n0} + \eta_{11} + \rho + \eta_{n2}), H_n\}$;
16) $\quad\quad$ If $k = 0$
17) $\quad\quad\quad k = n$;
18) $\quad\quad\quad$ Otherwise $k = k - 1$;
19) $\quad \Gamma^{j+1}_{n(n-2)} = \max\{(\Gamma^{j}_{n(n-1)} + \eta_{n1} + \rho + \eta_{(n-2)2}), H_{n-2}\}$;
20) $\quad$ If $n > 2$
21) $\quad\quad k = k - 1$;
22) $\quad\quad$ Otherwise $k = n$
23) $\quad j = j + 1$;
24) Stop;

By Algorithm 1, we follow the transition firing sequence to calculate the time delay. After firing s_{n1}, wafer W_{En} is loaded into a PM for processing at Step n modeled by p_n. Then, the robot goes to Step $n-2$ for unloading a wafer. Hence, $\Gamma^{1}_{n(n-2)} = \max\{\eta_{(n-2)2}, H_{n-2}\}$ (Line 1 in Algorithm 1) is the exact longest time delay for this process. Lines 3 and 4 lead to a different activity sequence for different n, i.e., $\langle s_{(n-2)2} \rightarrow s_{(n-1)1} \rightarrow y_{(n-3)1} \rightarrow y_{(n-3)2}\rangle$ if $n > 2$, and $\langle s_{(n-2)2} \rightarrow s_{(n-1)1} \rightarrow y_{n1} \rightarrow y_{n2}\rangle$ if $n = 2$. Then, via Lines 9, 13, and 15, Γ^{1}_{nk}, $\Gamma^{1}_{n(n-1)}$, and Γ^{1}_{nn} are obtained for different k. Based on the above results, $\Gamma^{2}_{n(n-2)}$ is obtained at Line 19. Continue this process, $\Gamma^{j}_{n(n-2)}$, $\Gamma^{j}_{n(n-3)}$, $\ldots$, Γ^{j}_{n0}, Γ^{j}_{nn}, $\Gamma^{j}_{n(n-1)}$, $\Gamma^{j+1}_{n(n-2)}$, $\ldots$, and $\Gamma^{m_n}_{n(n-2)}$ are obtained. Finally, when $\Gamma^{m_n}_{n0}$ is obtained at Line 9, the procedure stops.

If $m_n \leq m_u$, $u \in \mathbf{N_{n-1}}$, $\Gamma^{m_n}_{n0}$ can be obtained by Algorithm 1 such that the exact upper bound of B_n can be calculated by (10). If $m_i \le m_u$ and $i < n$, $u \in \mathbf{N_n}\backslash\{i\}$, to calculate B_i, we have to obtain $\Gamma^{m_i}_{i(i+1)}$. To do so, we need to renumber the steps as follows: 1) Step i as n; 2) Step j as Step $(j+n-i)$, $0 \le j < i$; 3) Step j as Step $(j-i-1)$, $i < j < n$; and 4) m_j, H_j, η_{j1}, and η_{j2} are numbered in the same way. In this way, B_i can be calculated just as B_n by using Algorithm 1 and (10).

Theorem 1: Assume that: 1) $m_n \le m_u$, $u \in \mathbf{N_{n-1}}$; and 2) $\Gamma^{m_n}_{n0}$ is obtained by Algorithm 1. Then, B_n given in (10) is the exact upper bound of the accumulated robot time delay when the robot arrives at q_{n3} again in the m_nth cycle after loading a wafer into Step n.

Proof: It is known that the robot tasks performed before loading a wafer into Step n have no effect on B_n, $n \in \mathbf{N_n}$. Thus, to calculate B_n, we need to consider the activity sequence that starts from loading a wafer into Step n only as done in Algorithm 1. Assume that after firing s_{n1}, wafer W_1 is loaded into Step n, and then the robot goes to Step $n-2$ by performing the robot task of $y_{(n-2)1}$ and waits in $q_{(n-2)3}$ for the completion of a wafer there. By scheduling, when the robot arrives at $q_{(n-2)3}$ under the normal conditions, there is a wafer completed with sojourn time Λ_{n-2}. With activity time variation, $\Gamma^1_{n(n-2)} = \max\{\eta_{(n-2)2}, H_{n-2}\}$ is the exact upper bound for this process. Then, after robot activity sequence $\kappa_4 = \langle s_{(n-2)2} \to$ waiting in $q_{(n-1)1} \to s_{(n-1)1} \to y_{(n-3)1} \to$ waiting in $q_{(n-3)2} \to y_{(n-3)2}\rangle$, the robot arrives at $q_{(n-3)3}$ for unloading a wafer there. With $\|\kappa_4\| = \eta_{(n-1)1} + \rho + \eta_{(n-3)2}$, when the robot arrives at $q_{(n-3)3}$, the longest delay is $\Gamma^1_{n(n-2)} + \eta_{(n-1)1} + \rho + \eta_{(n-3)2}$. Meanwhile, the longest delay caused by processing a wafer at Step $n-3$ is H_{n-3}. Thus, as done in Algorithm 1, $\Gamma^1_{n(n-3)} = \max\{\Gamma^1_{n(n-2)} + \eta_{(n-1)1} + \rho + \eta_{(n-3)2}, H_{n-3}\}$ is the exact upper bound. By Algorithm 1, every Γ^j_{nk} is calculated sequentially in this way, which guarantees that Γ^j_{nk} is the exact upper bound. After $\Gamma^{m_n}_{n0}$ is obtained, the robot comes back to q_{n3} for unloading wafer W_1. At this time, the longest time delay happens when W_1 is completed normally. It does not need to consider the time delay caused by its processing. Then, B_n is calculated according to (10) and it is the exact upper bound. ■

In [23] and [32], to obtain the time delay analytically, $H = \max\{H_1, \ldots, H_n\}$ is used as the delay in processing a wafer for all steps. Robot activities are similarly handled. Thus, they fail to obtain the exact upper bund. This problem is solved by Algorithm 1 in a sequential way. In Theorem 1, we consider just the situation that $m_n \le m_u$, $u \in \mathbf{N_{n-1}}$. However, if this condition is not true, Algorithm 1 is not applicable. Thus, we give Algorithm 2 for the case: $\exists f \ne n$ such that $m_n > m_f$.

To calculate B_n, consider the activity sequence that starts from loading wafer W_2 into Step n. For this case, if $f \ne n-2$, $f \ne n-1$, and $f \ne n$ hold, a wafer named as W_3 is loaded into Step f in the first cycle when transition s_{f1} fires. At this time, by Algorithm 1, the longest accumulated robot delay time is $\Gamma^1_{n(f-1)} + \eta_{f1} + \rho$. After m_f robot cycles, the robot goes to Step f for unloading W_3, or it arrives at q_{f3} in the (m_f+1)th robot cycle. By Algorithm 1, we have that $\Gamma^{m_f+1}_{nf} = \max\{\Gamma^{m_f+1}_{n(f+1)} + \eta_{(f+2)1} + \rho + \eta_{f2}, H_f\}$. However, note that, before loading W_3 into Step f, the longest robot delay is $\Gamma^1_{n(f-1)} + \eta_{f1} + \rho$. Thus, with the delay of the processing time at Step f considered, the accumulated delay time is $\Gamma^1_{n(f-1)} + \eta_{f1} + \rho + H_f$. Therefore, with the delay from both of wafer processing time at Step f and the robot task delay considered, we have $\Gamma^{m_f+1}_{nf} = \max\{\Gamma^1_{n(f-1)} + \eta_{f1} + \rho + H_f, \Gamma^{m_f+1}_{n(f+1)} + \eta_{(f+2)1} + \rho + \eta_{f2}\}$, when the robot leaves place q_{f3} in the (m_f+1)th cycle. Hence, Lines 10 and 11 in Algorithm 2 are used to calculate it. If $f = n-1$, the robot goes to place q_{f3} for unloading the wafer in the m_fth cycle which was loaded into Step f in the first cycle. Thus, with the processing time delay, $\Gamma^{m_f}_{n(n-1)} = \max\{\Gamma^1_{n(n-2)} + \eta_{(n-1)1} + \rho + H_{(n-1)}, \Gamma^{m_f}_{nn} + \eta_{01} + \rho + \eta_{(n-1)2}\}$. Lines 16 and 17 are used to calculate it. If $f = n-2$, the robot goes to place q_{f3} for unloading the wafer in the m_fth cycle that was loaded into Step f in the first cycle. The next robot activity is performed in the (m_f+1)th cycle. Therefore, with processing time delay, $\Gamma^{m_f+1}_{n(n-2)} =$

Algorithm 2: Calculate $\Gamma^{m_n}_{n0}$ when $\exists f \ne n$ such that $m_n > m_f$

If $\exists f \ne n$ such that $m_n > m_f$, calculate $\Gamma^{m_n}_{n0}$ as follows.

1) $\Gamma^1_{n(n-2)} = \max\{\eta_{(n-2)2}, H_{n-2}\}$;
2) If $i > 2$
3) $\quad k = n - 3$;
4) $\quad$Otherwise $k = n$;
5) $j = 1$;
6) While $j \le m_n$
7) $\quad$While $k \ne n-2$
8) $\quad\quad$If $0 \le k \le n-3$
9) $\quad\quad\quad \Gamma^j_{nk} = \max\{(\Gamma^j_{n(k+1)} + \eta_{(k+2)1} + \rho + \eta_{k2}), H_k\}$;
10) $\quad\quad$If $j = m_f + g$, $g \in \mathbf{N_n}$, and $k = f$
11) $\quad\quad\quad \Gamma^j_{nk} = \max\{\Gamma^{j-m_f}_{n(k-1)} + \eta_{k1} + \rho + H_k, \Gamma^j_{n(k+1)} + \eta_{(k+2)1} + \rho + \eta_{k2}\}$;
12) $\quad\quad$If $j = m_n$ and $k = 0$
13) $\quad\quad\quad$Go to Statement (30);
14) $\quad$If $k = n-1$
15) $\quad\quad \Gamma^j_{n(n-1)} = \max\{(\Gamma^j_{nn} + \eta_{01} + \rho + \eta_{(n-1)2}), H_{n-1}\}$;
16) $\quad$If $j = m_f + g$, $g \in \mathbf{\Omega}$, and $k = f$
17) $\quad\quad \Gamma^j_{n(n-1)} = \max\{\Gamma^{j-m_f+1}_{n(n-2)} + \eta_{(n-1)1} + \rho + H_{n-1}, \Gamma^j_{nn} + \eta_{01} + \rho + \eta_{(n-1)2}\}$;
18) $\quad$If $k = n$
19) $\quad\quad \Gamma^j_{nn} = \max\{(\Gamma^j_{i0} + \eta_{11} + \rho + \eta_{n2}), H_n\}$;
20) $\quad$If $k = 0$
21) $\quad\quad k = n$;
22) $\quad\quad$Otherwise $k = k - 1$;
23) $\quad \Gamma^{j+1}_{n(n-2)} = \max\{(\Gamma^j_{n(n-1)} + \eta_{n1} + \rho + \eta_{(n-2)2}), H_{n-2}\}$;
24) $\quad$If $j+1 = m_f + g$, $g \in \mathbf{N_n}$, and $f = n-2$
25) $\quad \Gamma^{j+1}_{n(n-2)} = \max\{\Gamma^{j+1-m_f}_{n(n-3)} + \eta_{(n-2)1} + \rho + H_{(n-2)}, \Gamma^j_{n(n-1)} + \eta_{n1} + \rho + \eta_{(n-2)2}\}$;
26) If $n > 2$
27) $\quad\quad k = k - 1$;
28) $\quad\quad$Otherwise $k = n$;
29) $j = j + 1$;
30) Stop;

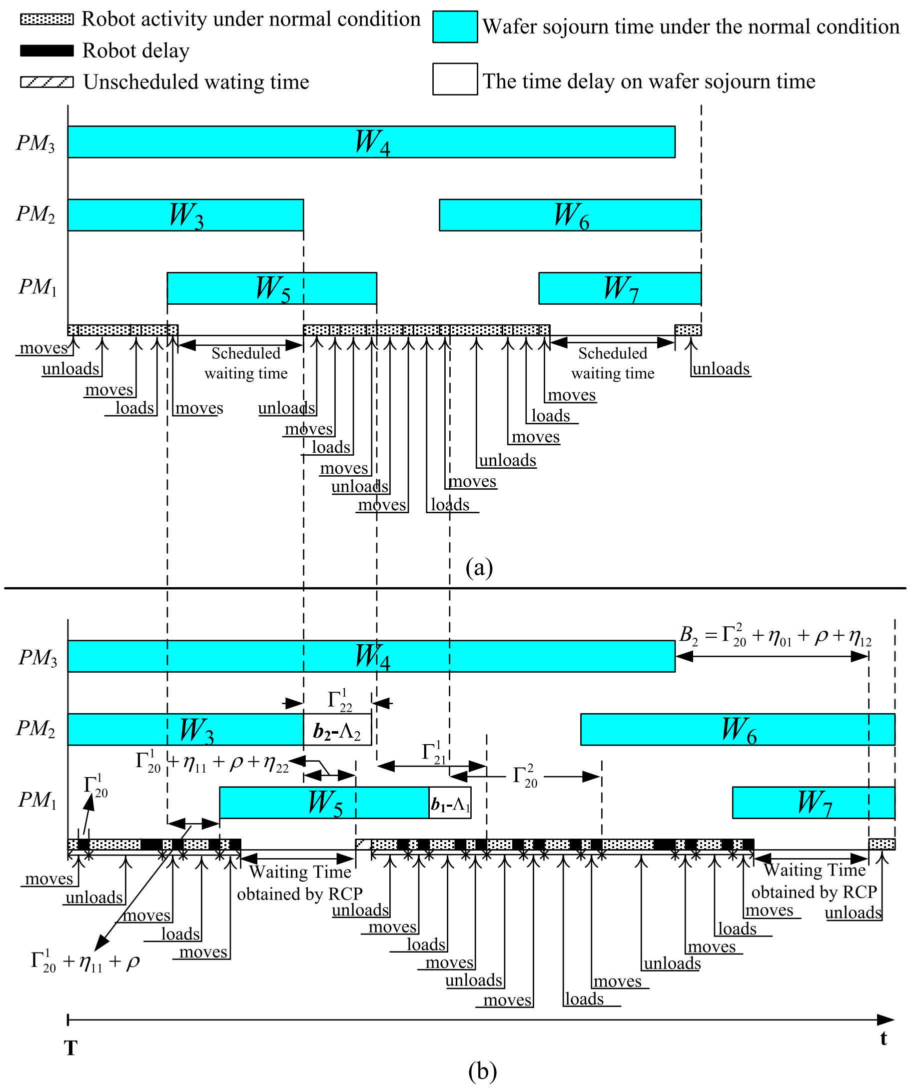

Fig. 3. Illustration of Algorithm 2 with wafer flow pattern (1, 2). (a) Schedule under the normal condition. (b) Schedule for the worst case by considering activity time variation.

$\max\{\Gamma^1_{n(n-3)} + \eta_{(n-2)1} + \rho + H_{(n-2)}, \Gamma^{m_f}_{n(n-1)} + \eta_{n1} + \rho + \eta_{(n-2)2}\}$, $n \geq 3$. Lines 24 and 25 are used to calculate it. The explanation of Algorithm 2 is shown in Fig. 3 via an example.

The example shown in Fig. 3 has two steps. PM_1 is used to process wafers at Step 1. PM_2 and PM_3 together are used to process wafers at Step 2. Thus, $m_1 = 1 < m_2 = 2$ holds. In this case, we explain how to obtain B_2. The accumulated robot delay at Step 2 is calculated when the robot starts from loading wafer W_4 into PM_3 as shown by the time point T in Fig. 3. Then, it moves to the loadlocks. At this time, delay Γ^1_{20} is obtained by Algorithm 2. The next robot task is $\langle s_{02} \rightarrow s_{11}\rangle$ such that wafer W_5 is loaded into PM_1. At this time, the accumulated time delay is $\Gamma^1_{20} + \eta_{11} + \rho$ as shown in Fig. 3(b). Then, y_{21} fires. Let $\kappa_5 = \langle s_{02} \rightarrow$ waiting in $q_{11} \rightarrow s_{11} \rightarrow y_{21} \rightarrow$ waiting in $q_{22} \rightarrow y_{22}\rangle$. With the RCP, we have $\|\kappa_5\| = \eta_{11} + \rho + \eta_{22}$. With $\Gamma^1_{22} = \max\{\Gamma^1_{20} + \eta_{11} + \rho + \eta_{22}, H_2\} = H_2$, when the robot arrives at PM_2, W_3 is not completed yet. Thus, the robot goes to q_{23} for an unscheduled waiting as shown in Fig. 3(b). Then, after W_3 is completed and $\langle s_{22} \rightarrow s_{01} \rightarrow y_{11}\rangle$ is performed, the robot goes to PM_1 for unloading W_5. Note that, W_5 is loaded into PM_1 after W_4 is loaded into PM_3. By Lines 16 and 17 in Algorithm 2, we have $\Gamma^1_{21} = \max\{\Gamma^1_{20} + \eta_{11} + \rho + H_1, \Gamma^1_{22} + \eta_{01} + \rho + \eta_{12}\}$. For this case, $\Gamma^1_{21} = \max\{\Gamma^1_{20} + \eta_{11} + \rho + H_1, \Gamma^1_{22} + \eta_{01} + \rho + \eta_{12}\} = \Gamma^1_{22} + \eta_{01} + \rho + \eta_{12}$ holds. Then, similar to Algorithm 1, we can obtain Γ^2_{20}. Thus, by (10), $B_2 = \Gamma^2_{20} + \eta_{01} + \rho + \eta_{12}$.

Similarly, with (10) and Algorithm 2, we can calculate B_i, $i \neq n$, by renumbering the steps and their corresponding parameters. In this way, Γ^j_{nk} can be calculated in a sequential way, which guarantees that Γ^j_{nk} is the exact upper bound. Thus, B_n given in (10) must be the exact upper bound of the accumulated robot time delay. Hence, we have the following theorem immediately.

Theorem 2: Assume that 1) $\exists f \neq n$ such that $m_n > m_f$; and 2) $\Gamma^{m_n}_{n0}$ is obtained via Algorithm 2. Then, B_n given in (10) is the exact upper bound of the accumulated robot time delay when the robot arrives at q_{n3} again in the m_nth cycle after loading a wafer into Step n.

Algorithm 3: Calculate $\Gamma_{n0}^{m_n}$ when $\exists f$ and $h, f \neq h \neq n$ such that $m_n > m_f$ and $m_f > m_h$

If $\exists f$ and $h, f \neq h \neq n$, such that $m_n > m_f$ and $m_f > m_h$, calculate $\Gamma_{n0}^{m_n}$ as follows.

1) $\Gamma_{n(n-2)}^{1} = \max\{\eta_{(n-2)2}, H_{n-2}\}$;
2) If $n > 2$
3) $k = n - 3$;
4) Otherwise $k = n$;
5) $j = 1$;
6) While $j \leq m_n$
7) While $k \neq n - 2$
8) If $0 \leq k \leq n - 3$
9) $\Gamma_{nk}^{j} = \max\{(\Gamma_{n(k+1)}^{j} + \eta_{(k+2)1} + \rho + \eta_{k2}), H_k\}$;
10) If $j = m_h + g$, $g \in \mathbf{N}_n$, and $k = h$
11) $\Gamma_{nk}^{j} = \max\{\Gamma_{n(k-1)}^{j-m_h} + \eta_{k1} + \rho + H_k, \Gamma_{n(k+1)}^{j} + \eta_{(k+2)1} + \rho + \eta_{k2}\}$;
12) If $j = m_f + g$, $g \in \mathbf{N}_n$, and $k = f$
13) $\Gamma_{nk}^{j} = \max\{\Gamma_{n(k-1)}^{j-m_f} + \eta_{k1} + \rho + H_k, \Gamma_{n(k+1)}^{j} + \eta_{(k+2)1} + \rho + \eta_{k2}\}$;
14) If $j = m_n$ and $k = 0$
15) Go to Statement (34);
16) If $k = n - 1$
17) $\Gamma_{n(n-1)}^{j} = \max\{(\Gamma_{nn}^{j} + \eta_{01} + \rho + \eta_{k2}), H_k\}$;
18) If $j = m_h + g$, $g \in \mathbf{\Omega}$, and $k = h$
19) $\Gamma_{n(n-1)}^{j} = \max\{\Gamma_{n(n-2)}^{j-m_h+1} + \eta_{(n-1)1} + \rho + H_{n-1}, \Gamma_{nn}^{j} + \eta_{01} + \rho + \eta_{(n-1)2}\}$;
20) If $j = m_f + g$, $g \in \mathbf{\Omega}$, and $k = f$
21) $\Gamma_{n(n-1)}^{j} = \max\{\Gamma_{n(n-2)}^{j-m_f+1} + \eta_{(n-1)1} + \rho + H_{n-1}, \Gamma_{nn}^{j} + \eta_{01} + \rho + \eta_{(n-1)2}\}$;
22) If $k = n$
23) $\Gamma_{nn}^{j} = \max\{(\Gamma_{n0}^{j} + \eta_{11} + \rho + \eta_{n2}), H_n\}$;
24) If $k = 0$
25) $k = n$;
26) Otherwise $k = k - 1$;
27) $\Gamma_{n(n-2)}^{j+1} = \max\{(\Gamma_{n(n-1)}^{j} + \eta_{n1} + \rho + \eta_{(n-2)2}), H_{n-2}\}$;
28) If $j + 1 = m_h + g$, $g \in \mathbf{N}_n$, and $h = n - 2$
29) $\Gamma_{n(n-2)}^{j+1} = \max\{\Gamma_{n(n-3)}^{j+1-m_h} + \eta_{(n-2)1} + \rho + H_{(n-2)}, \Gamma_{n(n-1)}^{j} + \eta_{n1} + \rho + \eta_{(n-2)2}\}$;
30) If $j + 1 = m_f + g$, $g \in \mathbf{N}_n$, and $f = n - 2$
31) $\Gamma_{n(n-2)}^{j+1} = \max\{\Gamma_{n(n-3)}^{j+1-m_f} + \eta_{(n-2)1} + \rho + H_{(n-2)}, \Gamma_{n(n-1)}^{j} + \eta_{n1} + \rho + \eta_{(n-2)2}\}$;
32) If $n > 2$
33) $k = k - 1$;
34) Otherwise $k = n$;
35) $j = j + 1$;
36) Stop;

Based on Algorithms 1 and 2, we have Algorithm 3 for the case: $\exists f$ and $h, f \neq h \neq n$, such that $m_n > m_f$ and $m_f > m_h$.

For this case, similar to Algorithm 2, every time the robot goes to q_{k3}, $k = f$ or h, for unloading the wafer that was loaded into Step k in or after the first cycle, Γ_{ik}^{j} can be calculated according to Lines 10–13, 18–21, and 28–31 in Algorithm 3. Similar to Algorithms 1 and 2, the calculation of B_i can be done by renumbering the steps and their parameters. Then, by this algorithm, every Γ_{nk}^{j} is calculated sequentially such that Γ_{nk}^{j} is the exact upper bound. Hence, based on Theorems 1 and 2, we have the following theorem.

Theorem 3: Assume that: 1) if $\exists f$ and $h, f \neq h \neq n$, such that $m_n > m_f$ and $m_f > m_h$; and 2) $\Gamma_{n0}^{m_n}$ is obtained via Algorithm 3. Then, B_n given in (10) is the exact upper bound of the accumulated robot time delay when the robot arrives at q_{n3} again in the m_nth cycle after loading a wafer into Step n.

Note that, our results can easily be extended to the case when $m_u \geq m_i > m_f > m_h > \cdots > m_e$, where $u \neq i \neq f \neq h \neq \cdots \neq e$. Thus, up to now, we present a method to calculate the exact upper bound of wafer sojourn time delay caused by activity time variation. By Algorithms 1–3, the number of the iteration times depends on the number of parallel PMs at a step and the number of steps. If there are m_n parallel PMs at Step n, we need to do the iteration for m_n cycles and, for each cycle, with n steps, we need to do it for n times. Thus, the computational complexity of the proposed method is $O(n \times m_n)$. In a cluster tool, both n and m_n are limited. Therefore, it is very efficient.

In [32], the upper bound of the wafer sojourn time delay is calculated by using analytical expressions. However, it is overestimated such that the schedulability conditions proposed in [23] are sufficient, but not necessary. By Algorithms 1–3, the exact upper bound for different cases can be obtained. Then, we can exactly check if a given off-line schedule is feasible. To make it feasible, i.e., making the PN model live, we require that $a_i \leq \tau_i \leq a_i + \delta_i$, $\forall i \in \mathbf{N}_n$. Thus, if a schedule is feasible under normal conditions, then, $a_i \leq \Lambda_i \leq a_i + \delta_i$ must hold. With the activity time variation considered, $\Lambda_i \leq \tau_i$ is always true. Thus, $a_i \leq \tau_i$ holds. Based on the presented results, we have $\tau_i \leq \Lambda_i + B_i$, where B_i obtained via Algorithms 1–3 is the exact upper bound of the wafer sojourn time at Step i. Hence, if, at the worst case, $\Lambda_i + B_i \leq a_i + \delta_i$ holds, the wafer residency constraints are never violated, or the schedule is feasible even if the activity time varies. With this perspective, by replacing the so-called upper bound of the wafer sojourn time in [23] by B_i calculated by Algorithms 1–3, the necessary and sufficient schedulability conditions are obtained. Thus, an optimal and feasible schedule can be found by using the scheduling algorithms presented in [23].

IV. Illustrative Examples

In this section, examples are used to show the applications and usefulness of the proposed approach.

Example 1: It is from [23] and the flow pattern is (1, 1). Under normal conditions, it takes 15 time units for the robot to unload a wafer from a loadlock and moves to Step 1 ($c_0 = 15$), and 10 time units for the robot to load a wafer into a PM or loadlock, or unload a wafer from a PM ($c = 10$), 2 time units to move from p_i to p_j ($\alpha = 2$). It needs 100 time units for a PM at both Steps 1 and 2 to process a wafer ($a_1 = a_2 = 100$), respectively. After being processed, a wafer at Steps 1 and 2 can stay there for no more than 20 time units ($\delta_1 = \delta_2 = 20$). The activity time is subject to random variation with $d_0 = 20$, $d = 12$, $\beta = 3$, and $b_1 = b_2 = 105$.

By applying the approach [23], it is obtained that $\omega_{11} = 0$, $\omega_{21} = 0$, $\omega_{02} = 0$, $\omega_{01} = 3$, $\omega_{12} = 1$, $\omega_{22} = 70$, $B_1 = 8$, and $B_2 = 9$. For this case, $m_1 = m_2$ holds. Thus, by the proposed method in this paper, Algorithm 1 and (10) are applied to obtain $B_1 = 6$ and $B_2 = 9$. It shows that, for B_2, the exact upper bound of the wafer sojourn time delay is obtained by the methods presented both in this paper and in [32]. However, B_1 is overestimated by 25% if the method in [32] is applied. This implies that, by the method proposed in this paper, a significant improvement is made. If a cluster tool is schedulable under normal conditions, we have $a_i \le \Lambda_i \le a_i + \delta_i$. With activity time variation, to check the feasibility, one needs to check if $\tau_i = \Lambda_i + B_i \le a_i + \delta_i$ holds. Therefore, overestimation of B_i may result in a feasible schedule being treated as an infeasible one. This situation can be completely avoided by the proposed method, which is further discussed via the next example.

Example 2: It is also from [23] and the flow pattern is (2, 2, 1). Under the normal conditions, $c_0 = 14$, $c = 10$, $\alpha = 2$, $a_1 = 150$, $a_2 = 140$, $a_3 = 48$, $\delta_1 = \delta_2 = 25$, and $\delta_3 = 20$. An activity time is subject to random variations, and we have $d_0 = 19$, $b_1 = 156$, $b_2 = 146$, and $b_3 = 53$.

Under the normal conditions, we have $\psi_1 = 2(n + 1)\alpha + (2n + 1)c + c_0 = 100$. By examining this case with the approach in [23], all the robot waiting times are set to be zero. Then, from (7)–(9), we have $\Lambda_1 = 150$, $\Lambda_2 = 154$, and $\Lambda_3 = 54$. Thus, with the activity time variation, by using the method in [32], one has $B_1 = 11$, $B_2 = 16$, and $B_3 = 11$. For this case, $m_1 = m_2 > \mathrm{m}_3$ holds. Thus, by Algorithm 2 and (10), we have $B_1 = B_2 = 11$. By Algorithm 1 and (10), we have $B_3 = 11$. B_2 is overestimated by 31.25% if the method in [32] is applied.

Next, we compare the schedule feasibility check via two methods. With the results obtained by using the approaches presented in [32], we have $B_1 + (\Lambda_1 - a_1) = 11 + (150 - 150) = 11 < \delta_1$, $B_2 + (\Lambda_2 - a_2) = 16 + (154 - 140) = 30 > \delta_2$, and $B_3 + (\Lambda_3 - a_3) = 11 + (54 - 48) = 17 < \delta_3$. In other words, for Step 2, the residency time constraints are violated. This implies that the schedule is infeasible. However, by the method presented in this paper, we have $B_1 + (\Lambda_1 - a_1) = 11 + (150 - 150) = 11 < \delta_1$, $B_2 + (\Lambda_2 - a_2) = 11 + (154 - 140) = 25 = \delta_2$, and $B_3 + (\Lambda_3 - a_3) = 11 + (54 - 48) = 17 < \delta_3$. This implies that the schedule is, in fact, feasible.

Example 3: The flow pattern is (1, 1, 1, 1, 1). Under normal conditions, $c_0 = 12$, $c = 8$, $\alpha = 3$, $a_1 = 90$, $a_2 = 80$, $a_3 = 95$, $a_4 = 90$, $a_5 = 90$, and $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = 25$. An activity time is subject to random variation, and we have $d_0 = 16$, $b_1 = 95$, $b_2 = 85$, $b_3 = 105$, $b_4 = 95$, and $b_5 = 95$.

Under normal conditions, we have $\psi_1 = 2(n + 1)\alpha + (2n + 1)c + c_0 = 136$. By examining this case with the approach in [23], all the robot waiting times are set to be zero. Then, from (7)–(9), we have $\Lambda_1 = 91$, and $\Lambda_2 = \Lambda_3 = \Lambda_4 = \Lambda_5 = 95$. Thus, with the activity time variation, by using the method in [32], one has $B_1 = 10$, $B_2 = 14$, $B_3 = 8$, $B_4 = 8$, and $B_5 = 14$. For this case, $m_1 = m_2 = m_3 = m_4 = m_5 = 1$ holds. Thus, by Algorithm 1 and (10) in this paper, we have $B_1 = 10$, $B_2 = 10$, $B_3 = 8$, $B_4 = 8$, and $B_5 = 14$. Therefore, B_2 is overestimated by about 28.6% if the method in [32] is applied.

Then, we compare the schedule feasibility check via two methods. With the results obtained by using the approach presented in [32], we have $B_1 + (\Lambda_1 - a_1) = 10 + (91 - 90) = 11 < \delta_1$, $B_2 + (\Lambda_2 - a_2) = 14 + (95 - 80) = 29 > \delta_2$, $B_3 + (\Lambda_3 - a_3) = 8 + (95 - 95) = 8 < \delta_3$, $B_4 + (\Lambda_4 - a_4) = 8 + (95 - 90) = 13 < \delta_4$, $B_5 + (\Lambda_5 - a_5) = 14 + (95 - 90) = 19 < \delta_5$. In other words, for Step 2, the residency time constraints are not satisfied. This implies that the schedule is infeasible. However, by the method presented in this paper, we have $B_1 + (\Lambda_1 - a_1) = 10 + (91 - 90) = 11 < \delta_1$, $B_2 + (\Lambda_2 - a_2) = 10 + (95 - 80) = 25 = \delta_2$, $B_3 + (\Lambda_3 - a_3) = 8 + (95 - 95) = 8 < \delta_3$, $B_4 + (\Lambda_4 - a_4) = 8 + (95 - 90) = 13 < \delta_4$, $B_5 + (\Lambda_5 - a_5) = 14 + (95 - 90) = 19 < \delta_5$. This implies that the schedule is, in fact, feasible.

V. Conclusion

Wafers in PMs in cluster tools face strict wafer residency time constraints. Such constraints greatly complicate the scheduling problem of cluster tools. Moreover, activity time variation may make a feasible schedule obtained under the deterministic activity time assumption infeasible. Thus, it is very challenging to operate a cluster tool with wafer residency time constraints and activity time variation. This problem is studied in [23], [25], [26], [31], and [32] for both single and dual-arm cluster tools. Based on a PN model and RCP, analytical expressions are derived to calculate the upper bound of wafer sojourn time delay in a PM. Then, real-time scheduling algorithms are proposed to find an optimal schedule. Nevertheless, the upper bound of wafer sojourn time delay is not the exact one but overestimated. Such overestimation may fail to identify some schedulable cases. To solve such a problem, this paper presents polynomial algorithms to find the exact upper bound of the wafer sojourn time delay. Based on it, one can check if a given schedule is feasible and find a feasible one if it is schedulable. The proposed method is of polynomial complexity.

For some wafer processing processes, a wafer needs to be processed in some PMs more than once, or there is wafer revisiting, which makes the scheduling problem more challenging. Our future work will deal with such cases.

References

[1] J.-H. Kim, T.-E. Lee, H.-Y. Lee, and D.-B. Park, "Scheduling analysis of timed-constrained dual-armed cluster tools," *IEEE Trans. Semicond. Manuf.*, vol. 16, no. 3, pp. 521–534, Aug. 2003.

[2] T.-E. Lee and S.-H. Park, "An extended event graph with negative places and tokens for timed window constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 2, no. 4, pp. 319–332, Oct. 2005.

[3] N. Q. Wu, C. B. Chu, F. Chu, and M. C. Zhou, "A Petri net method for schedulability and scheduling problems in single-arm cluster tools with wafer residency time constraints," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 2, pp. 224–237, May 2008.

[4] P. Burggraaf, "Coping with the high cost of wafer fabs," *Semicond. Int.*, vol. 18, no. 3, pp. 45–50, 1995.

[5] Y. Qiao, N. Q. Wu, and M. C. Zhou, "A Petri net-based novel scheduling approach and its cycle time analysis for dual-arm cluster tools with wafer revisiting," *IEEE Trans. Semicond. Manuf.*, vol. 26, no. 1, pp. 100–110, Feb. 2013.

[6] P. Singer, "The driving forces in cluster tool development," *Semicond. Int.*, vol. 18, no. 8, pp. 113–118, 1995.

[7] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Petri net modeling and cycle time analysis of dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 1, pp. 196–207, Jan. 2013.

[8] N. Q. Wu, M. C. Zhou, F. Chu, and C. B. Chu, "A Petri-net-based scheduling strategy for dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 5, pp. 1182–1194, Sep. 2013.

[9] W. K. V. Chan, J. Yi, and S. Ding, "Optimal scheduling of multicluster tools with constant robot moving times, part I: Two-cluster analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 5–16, Jan. 2011.

[10] S. Ding, J. Yi, and M. Zhang, "Scheduling multi-cluster tools: An integrated event graph and network model approach," *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 3, pp. 339–351, Aug. 2006.

[11] H.-J. Kim, J.-H. Lee, C. Jung, and T.-E. Lee, "Scheduling cluster tools with ready time constraints for consecutive small lots," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 1, pp. 145–159, Jan. 2013.

[12] T. L. Perkinson, P. K. MacLarty, R. S. Gyurcsik, and R. K. Cavin, III, "Single-wafer cluster tool performance: An analysis of throughput," *IEEE Trans. Semicond. Manuf.*, vol. 7, no. 3, pp. 369–373, Aug. 1994.

[13] T. L. Perkinson, R. S. Gyurcsik, and P. K. MacLarty, "Single-wafer cluster tool performance: An analysis of the effects of redundant chambers and revisitations sequences on throughput," *IEEE Trans. Semicond. Manuf.*, vol. 9, no. 3, pp. 384–400, Aug. 1996.

[14] S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, "A steady state throughput analysis of cluster tools: Dual-blade versus single-blade robots," *IEEE Trans. Semicond. Manuf.*, vol. 10, no. 4, pp. 418–424, Nov. 1997.

[15] N. Q. Wu and M. C. Zhou, "Colored timed Petri nets for modeling and analysis of cluster tools," *Asian J. Control*, vol. 12, no. 3, pp. 253–266, 2010.

[16] J. Yi, S. Ding, and M. Zhang, "Steady-state throughput and scheduling analysis of multi-cluster tools: A decomposition approach," *IEEE Trans. Autom. Sci. Eng*., vol. 5, no. 2, pp. 321–336, Apr. 2008.

[17] W. M. Zuberek, "Timed Petri nets in modeling and analysis of cluster tools," *IEEE Trans. Robot. Autom*., vol. 17, no. 5, pp. 562–575, Oct. 2001.

[18] Y.-H. Shin, T.-E. Lee, J.-H. Kim, and H.-Y. Lee, "Modeling and implementing a real-time scheduler for dual-armed cluster tools," *Comput. Ind.*, vol. 45, no. 1, pp. 13–27, 2001.

[19] N. Q. Wu and M. C. Zhou, *System Modeling and Control With Resource-Oriented Petri Nets*. New York, NY, USA: Taylor and Francis Group, Oct. 2009.

[20] D. Jevtic, "Method and apparatus for priority based scheduling of wafer processing within a multiple chamber semiconductor wafer processing tool," U.S. patent 5 928 389, 1999.

[21] T.-E. Lee, H.-Y. Lee, and Y.-H. Shin, "Workload balancing and scheduling of a single-armed cluster tool," in *Proc. 5th Asia Pacific Ind. Eng. Manage. Syst. Conf. (APIEMS)*, Gold Coast, QLD, Australia, 2004, pp. 1–15.

[22] M.-J. Lopez and S.-C. Wood, "Systems of multiple cluster tools—Configuration, reliability, and performance," *IEEE Trans. Semicond. Manuf.*, vol. 16, no. 2, pp. 170–178, May 2003.

[23] Y. Qiao, N. Q. Wu, and M. C. Zhou, "Real-time scheduling of single-arm cluster tools subject to residency time constraints and bounded activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 564–577, Jul. 2012.

[24] S. Rostami, B. Hamidzadeh, and D. Camporese, "An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 609–618, Oct. 2001.

[25] N. Q. Wu and M. C. Zhou, "Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 446–454, Apr. 2012.

[26] N. Q. Wu and M. C. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012.

[27] H. J. Yoon and D. Y. Lee, "On-line scheduling of integrated single-wafer processing tools with temporal constraints," *IEEE Trans. Semicond. Manuf.*, vol. 18, no. 3, pp. 390–398, Aug. 2005.

[28] N. Q. Wu and M. C. Zhou, "A closed-form solution for schedulability and optimal scheduling of dual-arm cluster tools with wafer residency time constraint based on steady schedule analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 2, pp. 303–315, Apr. 2010.

[29] J.-H. Kim and T.-E. Lee, "Schedule stabilization and robust timing control for timed-constrained cluster tools," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, Sep. 2003, pp. 1039–1044.

[30] J.-H. Kim and T.-E. Lee, "Schedulability analysis of timed-constrained cluster tools with bounded time variation by an extended Petri net," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 3, pp. 490–503, Jul. 2008.

[31] N. Q. Wu and M. C. Zhou, "Analysis of wafer sojourn time in dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Semicond. Manuf.*, vol. 23, no. 1, pp. 53–64, Feb. 2010.

[32] Y. Qiao, N. Q. Wu, and M. C. Zhou, "Petri net modeling and wafer sojourn time analysis of single-arm cluster tools with residency time constraints and activity time variation," *IEEE Trans. Semicond. Manuf.*, vol. 25, no. 3, pp. 432–446, Aug. 2012.

[33] Y. Qiao, N. Q. Wu, and M. C. Zhou, "Real-time control policy for single-arm cluster tools with residency time constraints and activity time variation by using Petri net," in *Proc. IEEE Int. Conf. Netw. Sens. Control*, Beijing, China, 2012, pp. 34–39.

[34] Y. Qiao, N. Q. Wu, and M. C. Zhou, "Petri net-based real-time scheduling of time-constrained single-arm cluster tools with activity time variation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, 2012, pp. 5056–5061.

[35] Y. Qiao, C. R. Pan, N. Q. Wu, and M. C. Zhou, "Response policies to process module failure in single-arm cluster tools subject to wafer residency time constraints," *IEEE Trans. Autom. Sci. Eng.*, to be published.

[36] Y. Qiao, N. Q. Wu, P. C. Rong, and M. C. Zhou, "How to respond to process module failure in residency time-constrained single-arm cluster tools," *IEEE Trans. Semicond. Manuf.*, vol. 27, no. 4, pp. 462–474, Nov. 2014.

[37] Y. Qiao, N. Q. Wu, Q. H. Zhu, and L. P. Bai, "Cycle time analysis of dual-arm cluster tools for wafer fabrication processes with multiple wafer revisiting times," *Comput. Oper. Res.*, to be published.

[38] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Petri net-based scheduling of single-arm cluster tools with reentrant atomic layer deposition processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 42–55, Jan. 2011.

[39] Y. Qiao, N. Q. Wu, and M. C. Zhou, "Scheduling of dual-arm cluster tools with wafer revisiting and residency time constraints," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 286–300, Feb. 2014.

[40] D.-Y. Liao, M. D. Jeng, and M. C. Zhou, "Petri net modeling and Lagrangian relaxation approach to vehicle scheduling in 300 mm semiconductor manufacturing," in *Proc. 2004 IEEE Conf. Robot. Autom.*, New Orleans, LA, USA, pp. 5301–5306.

[41] N. Q. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.,* vol. 29, no. 2, pp. 192–204, May 1999.

[42] N. Q. Wu and M. C. Zhou, "Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 658–669, Oct. 2001.

[43] N. Q. Wu and M. C. Zhou, "Deadlock avoidance in semiconductor track systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, USA, May 2002, pp. 193–198.

[44] N. Q. Wu and M. C. Zhou, "Modeling and deadlock control of automated guided vehicle systems," *IEEE/ASME Trans. Mechatron.*, vol. 9, no. 1, pp. 50–57, Mar. 2004.

[45] N. Q. Wu and M. C. Zhou, "Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1193–1202, Dec. 2005.

[46] N. Q. Wu and M. C. Zhou, "Real-time deadlock-free scheduling for semiconductor track systems based on colored timed Petri nets," *OR Spectr.*, vol. 29, no. 3, pp. 421–443, 2007.

[47] M. Zhou and F. DiCesare, "Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 515–527, Aug. 1991.

[48] M. Zhou, F. DiCesare, and A. Desrochers, "A hybrid methodology for synthesis of Petri nets for manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 350–361, Jun. 1992.

[49] M. C. Zhou and M. D. Jeng, "Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 3, pp. 333–357, Aug. 1998.

[50] N. Q. Wu and M. C. Zhou, "Deadlock resolution in automated manufacturing systems with robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 3, pp. 474–480, Jul. 2007.

[51] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Petri net-based optimal one-wafer cyclic scheduling of hybrid multi-cluster tools in wafer fabrication," *IEEE Trans. Semicond. Manuf.*, vol. 27, no. 2, pp. 192–203, May 2014.

[52] Q. H. Zhu and Y. Qiao, "Scheduling single-arm multi-cluster tools with lower bound cycle time via Petri nets," *Int. J. Intell. Control Syst.*, vol. 17, no. 4, pp. 113–123, 2012.

[53] Q. H. Zhu, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Petri net-based optimal one-wafer scheduling of single-arm multi-cluster tools in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 26, no. 4, pp. 578–591, Nov. 2013.

[54] M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: Petri Net Approach*. Singapore: World Scientific, 1998.

[55] J.-H. Kim, M. C. Zhou, and T.-E. Lee, "Schedule restoration for single-armed cluster tools," *IEEE Trans. Semicond. Manuf.*, vol. 27, no. 3, pp. 388–399, Aug. 2014.

[56] Z. H. Ding, M. C. Zhou, and S. G. Wang, "Ordinary differential equation based deadlock detection," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 10, pp. 1435–1454, Oct. 2014.

[57] Y. Chen, Z. Li, and M. C. Zhou, "Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 549–563, Apr. 2014.

[58] Z. W. Li, G. Y. Liu, M.-H. Hanisch, and M. C. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.

ChunRong Pan received the M.S. degree in mechatronics engineering from Shantou University, Shantou, China, and the Ph.D. degree in industrial engineering from the Guangdong University of Technology, Guangzhou, China, in 2006 and 2010, respectively.

From 1997 to 2011, he was with Shantou University. He joined the Jiangxi University of Science and Technology, Ganzhou, China, in 2011, where he is currently an Associate Professor with the Department of Mechatronics Engineering. He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA from 2013 to 2014. His current research interests include modeling, simulation, scheduling, and control of integrated manufacturing equipment.

Yan Qiao received the B.S. degree in industrial engineering from the Guangdong University of Technology, Guangzhou, China, in 2009. Since 2009, he has been pursuing the Ph.D. degree in industrial engineering from the Department of Industrial Engineering, Guangdong University of Technology.

His current research interests include discrete event systems, production planning, Petri nets, scheduling, and control.

Mr. Qiao was the recipient of the 2011 IEEE International Conference on Automation Science and Engineering QUALTECH System INC. Best Application Paper Award Finalist and the Best Student Paper Award in IEEE International Conference on Networking, Sensing and Control in 2012.

NaiQi Wu (M'04–SM'05) received the B.S. degree in electrical engineering from the Huainan Institute of Technology, Huainan, China, in 1982, and the M.S. and Ph.D. degrees in systems engineering, both from Xi'an Jiaotong University, Xi'an, China, in 1985 and 1988, respectively.

From 1988 to 1995, he was with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, and from 1995 to 1998, with Shantou University, Shantou, China. He moved to the Guangdong University of Technology, Guangzhou, China, in 1998, and joined the Macau University of Science and Technology, Macau, China, in 2013. From 1991 to 1992, he was a Visiting Scholar with the School of Industrial Engineering, Purdue University, West Lafayette, IN, USA. In 1999, 2004, and from 2007 to 2009, he was a Visiting Professor with the Department of Industrial Engineering, Arizona State University, Tempe, AZ, USA, the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA, and the Industrial Systems Engineering Department, the Industrial Systems Optimization Laboratory, University of Technology of Troyes, Troyes, France, respectively. He is currently a Professor with the Institute of Systems Engineering, Macau University of Science and Technology. His current research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri nets, intelligent transportation systems, and information assurance. He has authored or coauthored one book, four book chapters, and over 50 papers in international journals.

Dr. Wu was an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART-C and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and an Editor-in-Chief of the *Industrial Engineering Journal*. He has served for several international conferences as a Program Committee Member, the Program Committee Co-Chair of the 2012 IEEE International Conference on Networking, Sensing, and Control, and the Program Committee Chair of the 2013 IEEE International Conference on Networking, Sensing, and Control.

MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1983, 1986, and 1990, respectively.

He joined the New Jersey Institute of Technology, Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor of Electrical and Computer Engineering. He is also a Professor with Tongji University, Shanghai, China. His current research interests include Petri nets, sensor networks, web services, big data, semiconductor manufacturing, transportation, and energy systems. He has over 560 publications including 11 books, 270+ journal papers (majority in IEEE Transactions), and 22 book-chapters.

Prof. Zhou is the Founding Editor of the *IEEE Press Book Series on Systems, Science, and Engineering*, and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He is a Life Member of the Chinese Association for Science and Technology-USA and served as its President, in 1999. He is a Fellow of the International Federation of Automatic Control and American Association for the Advancement of Science.