This document is downloaded from DR-NTU (https://dr.ntu.edu.sg)
Nanyang Technological University, Singapore.

Efficient approach to scheduling of transient processes for time-constrained single-arm cluster tools with parallel chambers

Yang, Fajun; Qiao, Yan; Gao, Kaizhou; Wu, Naiqi; Zhu, Yuting; Ware, Simon Ian; Su, Rong

2020

Yang, F., Qiao, Y., Gao, K., Wu, N., Zhu, Y., Ware, S. I., & Su, R. (2018). Efficient approach to scheduling of transient processes for time-constrained single-arm cluster tools with parallel chambers. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50(10), 3646-3657. doi:10.1109/tsmc.2018.2852724

https://hdl.handle.net/10356/143809

https://doi.org/10.1109/TSMC.2018.2852724

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: https://doi.org/10.1109/TSMC.2018.2852724

Downloaded on 26 Mar 2023 05:16:28 SGT

Efficient Approach to Scheduling of Transient Processes for Time-Constrained Single-Arm Cluster Tools with Parallel Chambers

FaJun Yang, Yan Qiao, *Member*, *IEEE*, KaiZhou Gao, *Member*, *IEEE*, NaiQi Wu, *Senior Member*, *IEEE*, YuTing Zhu, Ian Ware Simon, Rong Su, *Senior Member*, *IEEE*

***Abstract*-In wafer manufacturing, extensive research on the operations of cluster tools under the steady state has been reported. However, with the shrinking down of wafer lot size, such tools are frequently required to switch from handling one lot of wafers to another, resulting in more transient processes, including start-up and close-down processes. Also, wafer residency time constraint is critical for many wafer fabrication processes. To address the transient scheduling problem of time-constrained single-arm cluster tools with parallel chambers, based on a generalized backward strategy, this work first builds timed Petri net models for the two transient processes. Then, two linear programs are derived for the first time to search a feasible schedule with minimal makespan. Two industrial examples are given to demonstrate the effectiveness of the obtained results at last.**

Index Terms—**Cluster tools, wafer manufacturing, transient process, Petri net, scheduling.**

I. Introduction

As a kind of computer-controlled complex equipment, cluster tools are extensively used for processing wafers in semiconductor fabrication. Generally, they are formed by several process chambers (PCs), two loadlocks, and a robot. The robot located at the center of the tool could be single or dual-arm one, resulting in single and dual-arm cluster tools. In Fig. 1, the architecture of a single-arm cluster tool is illustrated. Through the loadlocks, raw wafers to be processed enter the tool in a cassette-by-cassette way. Then, the robot is instructed to pick up a wafer from a cassette and transport it to PCs for processing in a predetermined sequence. After visiting all processing steps, a completed wafer returns to the cassette [37].

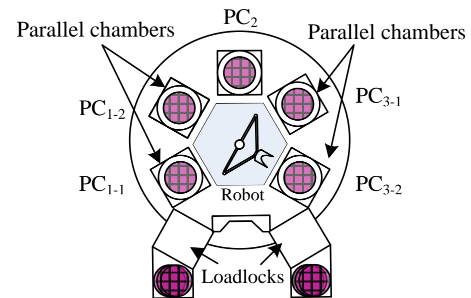


Fig. 1. (a) single-arm cluster tool.

Wafer residency time constraint (WRTC) is critical for many wafer fabrication processes, which requires that a processed wafer in a PC can stay there for no more than a given time and its violation could result in a quality problem [7, 28]. A cluster tool with such a constraint is called a time-constrained one.

In comparison with the scheduling problem of complex industrial processes [4, 33, 40, 41, 49], it seems that scheduling a cluster tool is relatively simple. However, due to WRTC and that it needs to schedule the robot tasks and wafer processing simultaneously, it is challenging to schedule a cluster tool. Thus, during the last two decades, much effort has been done for scheduling cluster tools [1, 7, 15, 28, 30, 32, 37-39, 42-44, 52]. In [15], methods for finding a feasible schedule for both single and dual-arm cluster tools with WRTC are derived. The drawback is that the proposed methods involve complex computation. In [30, 37-39], for time-constrained single and dual-arm cluster tools, an optimal and feasible schedule can be very efficiently obtained by their derived analytical expressions if the schedulability conditions are satisfied. In [44], two linear programs are developed to find a feasible schedule for single-arm cluster tools if the schedulability conditions are satisfied. In comparison with [30] where the scheduling strategy is limited to be backward, the strategy in [44] can be arbitrary.

For single-arm multi-cluster tools with WRTC, Zhu *et al.* [52] develop schedulability conditions for the existence of a

This work was supported in part by the Delta Electronics Inc and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme, in part by Science and Technology development fund (FDCT) of Macau under Grant 106/2016/A3, in part by the National Natural Science Foundation of China under Grant U1401240. (Corresponding authors: Naiqi Wu and Fajun Yang)

F. J. Yang, K. Z. Gao, Y. T. Zhu, I. W. Simon, and R. Su, and are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore, Singapore (e-mail: fjyang@ntu.edu.sg, kzgao@ntu.edu.sg, YUTING002@e.ntu.edu.sg, sware@ntu.edu.sg, and RSu@ntu.edu.sg).

Y. Qiao is with the Institute of Systems Engineering, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau (e-mail: yqiao@must.edu.mo).

N. Q. Wu is with the Institute of Systems Engineering, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau, and also the national key laboratory of precise electronic manufacturing technology and equipment, Guangdong University of Technology, Guangzhou 510006, China (e-mail: nqwu@must.edu.mo).

feasible schedule and algorithms to find an optimal one if the conditions are met. Yang *et al.* [42] extend the result to hybrid multi-cluster tools that consist of both single and dual-arm cluster tools.

Due to product customization and shrinking down of wafer circuit line width, the wafer lot size is becoming smaller and smaller. Hence, a cluster tool often needs to switch from handling one lot of wafers to another, resulting in more transient processes [9, 11-13, 26, 53]. Also, corrective and preventive maintenance could lead to unexpected close-down of a tool, which is often accompanied by the emergence of transient processes. Thus, scheduling a transient process becomes an important issue to be addressed.

For lot switching, Lee *et al*. [12] obtain an optimal robot sequence with minimal makespan by using an MIP model. Given the robot sequences, Lee *et al*. [13] derive closed-form expressions for such a process to minimize the makespan for both single and dual-arm cluster tools. Kim *et al*., [9] propose a branch and bound procedure to minimize the makespan. In [11], by addressing single-arm cluster tools with parallel chambers, the authors propose a generalized backward strategy in an incremental and decremental way for start-up and close-down processes, respectively. It is proved that such a generalized backward strategy can obtain an optimal schedule in most cases in terms of makespan. However, in their work, no WRTC is taken into account.

In [26, 53], by developing a Petri Net (PN) model to describe the characteristics of single-arm cluster tools, scheduling algorithms are developed to search for an optimal and feasible schedule for the start-up and close-down processes with a WRTC. However, in their work, only one PC is configured for each processing step. A comprehensive survey on cluster tool scheduling can be found in [23].

In practice, to balance the workloads, multiple PCs need to be configured in parallel for a processing step. With parallel chambers at a step, in the transient processes, as the wafers processed in different PCs for the same step are unloaded during different robot cycles whose cycle times may not be identical, hence, the wafer residency time at the same step may also be different. Thus, in comparison with [26, 53], the addressed situations in this work are more practical and challenging, since parallel chambers are taken into consideration. As pointed out by Wu *et al*. [30], given a cyclic schedule under a steady state, when the system reaches the steady state via a start-up transient process, this state should be compatible with the given cyclic schedule in terms of the robot task scheduling; otherwise, the cyclic schedule cannot be implemented. This is also challenging to schedule a transient process, especially when there are parallel PCs at a step.

To obtain an optimal and feasible schedule for a transient period based on a generalized backward strategy, this work establishes two PN models to describe the characteristic of system for the two transient processes. Consequently, for the first time, it derives two linear programs to obtain a feasible schedule with minimal makespan for the start-up and close-down processes. Note that a linear program can be solved by a polynomial algorithm. Furthermore, with no more than six chambers in a cluster tool, the size of the developed linear programs cannot be large. Thus, a solution can be very efficiently found, which is significant in scheduling a cluster tool.

In the next Section, two timed PN models are developed for start-up and close-down processes, respectively. Based on them, Sections III and IV derive two linear programs to obtain an optimal and feasible schedule for the start-up and close-down processes, respectively. Examples are used to demonstrate the effectiveness of the proposed approach in Section V. Finally, conclusions are summarized in Section VI.

II. System Modeling and Control

Let $\mathbb{N}_n = \{1, 2, \ldots, n\}$ and $\boldsymbol{\Omega}_n = \mathbb{N}_n \cup \{0\}$. For a single-arm cluster tool, we assume that it consists of n processing steps (except the loadlocks) and m_i ($m_i \geq 1$) parallel PCs are configured for Step i. Thus, a wafer flow pattern can be indicated as (m_1, m_2, …, m_n). For concise presentation, the loadlocks are treated as both Steps 0 and n + 1 with no processing function.

For a single-arm cluster tool in the steady state, a backward strategy is optimal [17] due to that, in most cases, the wafer processing time in a PC is much longer than the robot activity time. For the start-up and close-down processes, Kim *et al*. [11] prove that a generalized backward strategy can obtain a schedule to achieve a minimal makespan in most cases. Furthermore, due to its simplicity and efficiency, such a generalized backward strategy is widely used for single-arm cluster tools [14]. Hence, such strategies are adopted for the steady state and transient process in this work. As backward strategy is well known in this field, we just explain how it is generalized for the start-up and close-down processes in the follows.

A. Generalized Backward Strategy for Transient Process

For the start-up process, let M_d, $d \in \mathbb{N}_n$, denote the state at which there are m_i wafers being processing at Step i, $i \in \mathbb{N}_d$, while Step j, $d < j \leq n$, is idle. M_0 represents the initial state at which each PC is empty.

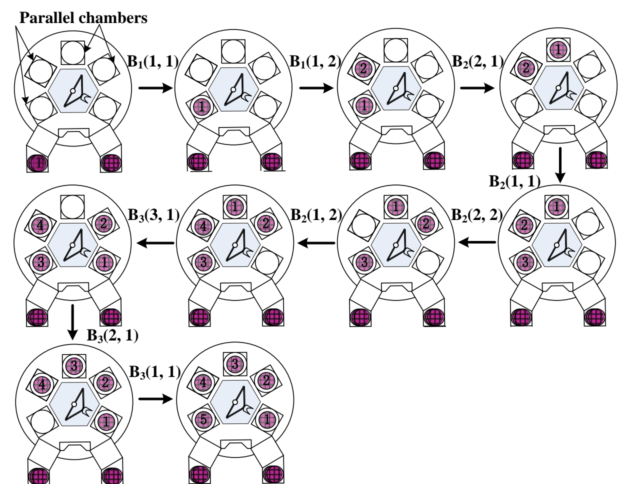


Fig. 2. The illustration of a generalized backward strategy for the start-up process.

Define a basic backward operation A_i performed by the robot at Step i with A_i = ⟨unloading a wafer from Step i - 1 → moving to Step i → loading it into Step i⟩, $i \in \mathbb{N}_{n+1}$. Further,

define $B_d(i, j)$ as the jth basic backward operation A_i at Step i during the evolution process from state M_{d-1} to M_d, $d \in \mathbb{N}_n$. For the steps with parallel chambers, wafers in them follow a "first-in-first-out" rule. Thus, the generalized backward strategy for the start-up process can be denoted as:

1) $B_1(1, 1) \rightarrow B_1(1, 2) \rightarrow \ldots \rightarrow B_1(1, m_1)$;
2) $B_d(d, 1) \rightarrow B_d(d\text{-}1, 1) \rightarrow \ldots \rightarrow B_d(1, 1) \rightarrow B_d(d, 2) \rightarrow \ldots \rightarrow B_d(d, m_d) \rightarrow B_d(d\text{-}1, m_d) \rightarrow \ldots \rightarrow B_d(1, m_d)$, $2 \leq d \leq n$.

After performing $B_n(1, m_n)$, for each PC in the tool, there is a wafer that is being processed, i.e., the start-up process is completed. The start-up process of a cluster tool with wafer flow pattern (2, 2, 1) is illustrated in Fig. 2.

For the close-down process, let W_d, $d \in \mathbb{N}_n$, denote the state at which there are m_i wafers being processing at Step i, $d \leq i \leq n$, while Step j, $1 \leq j < d$, is idle. Further, define $C_d(i, j)$ as the jth basic backward operation A_i at Step i during the state evolution process from W_d to W_{d+1}, $d \in \mathbb{N}_n$, where W_1 represents the state at which there is a wafer in processing in each PC and W_{n+1} represents that all PCs are emptied. With a "first-in-first-out" rule, the generalized backward strategy for the close-down process can be denoted as:

1) $C_d(n+1, 1) \rightarrow C_d(n, 1) \rightarrow C_d(n\text{-}1, 1) \rightarrow \ldots \rightarrow C_d(d+1, 1) \rightarrow C_d(n+1, 2) \rightarrow \ldots \rightarrow C_d(n+1, m_d) \rightarrow C_d(n, m_d) \rightarrow C_d(n\text{-}1, m_d) \rightarrow \ldots \rightarrow C_d(d+1, m_d)$, $1 \leq d \leq n\text{-}1$;
2) $C_n(n+1, 1) \rightarrow C_n(n+1, 2) \rightarrow \ldots \rightarrow C_n(n+1, m_n)$.

After performing $C_n(n+1, m_n)$, the last wafer is completed and loaded into a loadlock such that the tool goes to the initial idle state. The close-down process of a cluster tool with wafer flow pattern (2, 2, 1) is illustrated in Fig. 3.

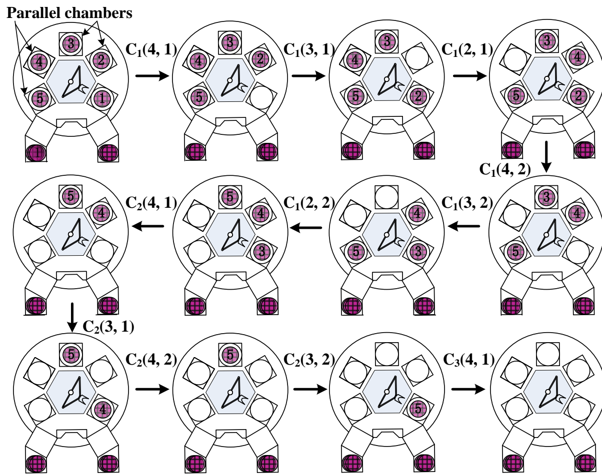


Fig. 3. The illustration of a generalized backward strategy for the close-down process.

B. Finite Capacity Petri Net

As an efficient tool for modeling, analysis, and control, PNs are widely applied in discrete manufacturing systems [2, 5-10, 14, 18-22, 24, 25, 27, 30-32, 45-50]. By following the work in [29, 30, 34-36, 51], the definition of a finite capacity PN is given as follows.

Definition 1: A finite capacity PN is a six-tuple digraph with the following set of elements.

1) $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places.
2) $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions, where $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.
3) I: $(P \times T) \rightarrow \mathbf{N} = \{0, 1, 2, \ldots\}$ is an input function specifying the existence of a directed arc from place p to transition t with weight $I(p, t)$.
4) O: $(P \times T) \rightarrow \mathbf{N}$ is an output function specifying the existence of a directed arc from t to p with weight $O(p, t)$.
5) $M(P) = (M(p_1), \ldots, M(p_m))^{\mathrm{T}}$ is a marking, where $M(p_i)$ represents the number of tokens in p_i.
6) $\mathcal{K}$: a capacity function with $\mathcal{K}(p)$ denoting the maximum number of tokens that p can accommodate.

Note that $I(p, t) = 0$ or $O(p, t) = 0$ implies that arc (p, t) or (t, p) does not exist. Usually, places and transitions in a PN model are denoted by circles and bars. Details about the transition enabling and firing rules of PNs can be found in [29, 34, 36]. Next, we use the finite capacity PN to model the system.

C. Modeling the Steady State Process

With a backward strategy, the PN model for a single-arm tool under the steady state is developed in [44] and it is depicted in Fig. 4.

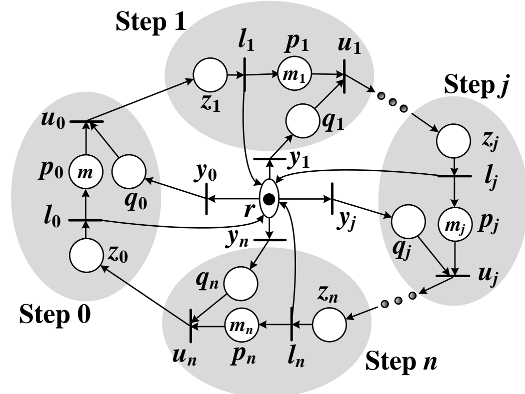


Fig. 4. The PN model for a single-arm cluster tool under the steady-state.

In Fig. 4, p_0 models the loadlocks with $\mathcal{K}(p_0) = m$ representing that there are always wafers to be processed in the loadlocks. For Step j, $j \in \mathbb{N}_n$, place p_j with $\mathcal{K}(p_j) = m_j$ models the step and indicates that there are m_j PCs. For each Step j, $j \in \mathbf{\Omega}_n$, place q_j with $\mathcal{K}(q_j) = 1$ models that the robot waits there and ready to remove a processed wafer from one of the PCs. Place z_j, $j \in \mathbf{\Omega}_n$ with $\mathcal{K}(z_j) = 1$, represents that the robot rotates from Steps j-1 to j with a wafer being held. Note that both Steps 0 and $n + 1$ model the loadlocks. Hence, z_0 models the robot moving from Steps n to 0 or $n + 1$ with a wafer being held. The robot is modeled by place r with $\mathcal{K}(r) = 1$, representing that there is one robot arm only.

The loading/unloading activity of the robot at Step j, $j \in \mathbf{\Omega}_n$, is modelled by transition l_j/u_j. The robot moving from a step to Step j with no wafer being held is modeled by y_j, $j \in \mathbf{\Omega}_n$. Note that y_j, $j \in \mathbb{N}_n$, can also denote the robot moving from one chamber to another at Step j with no wafer being held. Then, by adding arcs $(u_{j\text{-}1}, z_j)$, (z_j, l_j), (l_j, p_j), (p_j, u_j), (l_i, r), (r, y_j), (y_j, q_j), and (q_j, u_j), $j \in \mathbf{\Omega}_n$, we can obtain the PN structure for the system under the steady state as shown in Fig. 4.

For the start-up process, when the tool needs to start for operation, the robot acts according to the following task sequence: ⟨unloading wafer 1 from a loadlock → moving to

Step 1 → loading it there → moving back to the loadlock if $m_1 \geq 2$, otherwise, state M_1 is reached → unloading wafer 2 from the loadlock if $m_1 \geq 2$ → moving to Step 1 → loading it there → … → unloading wafer m_1 from the loadlock → moving to Step 1 → loading it there⟩ such that state M_1 is reached.

During the evolution process from M_{d-1} to M_d, $d \in \mathbb{N}_n \backslash \{1\}$, the robot follows the following task sequence: ⟨moving to Step d - 1 → unloading a wafer there → moving to Step d → loading the wafer there → moving to Step d - 2 → unloading a wafer there → moving to Step d - 1 → loading the wafer there → … → moving to a loadlock → unloading a wafer there → moving to Step 1 → loading the wafer there⟩. By repeating this process m_d times, M_d is reached. Finally, when M_n is reached, the start-up process is completed and the steady state is reached. Hence, the start-up process is defined as the process from M_0 to M_n. A PN model for the start-up process from state M_{d-1} to M_d, $d \in \mathbb{N}_n$, is shown in Fig. 5, where the meaning of the places and transitions is identical to that in Fig. 4. Note that, in Fig. 5, place p_d modeling Step d is marked by k tokens with $0 \leq k < m_d$.

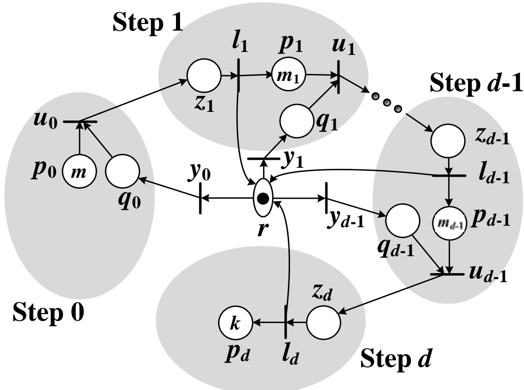


Fig. 5. The PN model for the start-up process of a single-arm cluster tool from state M_{d-1} to M_d, $d \in \mathbb{N}_n$.

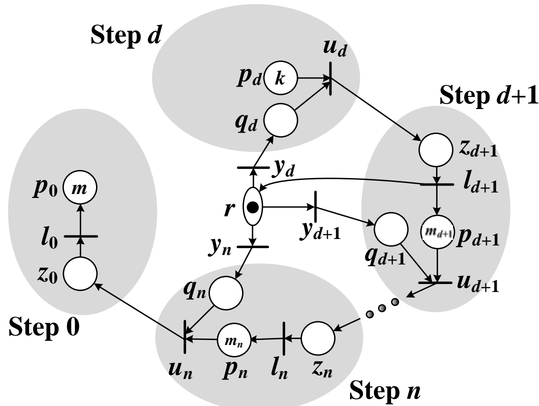


Fig. 6. The PN model for the close-down process of a single-arm cluster tool from state W_d to W_{d+1}, $d \in \mathbb{N}_n$.

Next, for the close-down process, during the process from state W_d to W_{d+1}, $d \in \mathbb{N}_{n-1}$, the robot acts according to the following task sequence: ⟨moving to Step n → unloading a wafer from it → moving to a loadlock → loading the wafer there → moving to Step n - 1 → unloading a wafer there → moving to Step n → loading the wafer there → … → moving to Step d → unloading a wafer there → moving to Step d + 1 → loading the wafer there⟩. By repeating this process m_d times, state W_{d+1} is reached.

From state W_n to W_{n+1}, the robot follows the following task sequence: ⟨unloading a wafer from Step n → moving to a loadlock → loading the wafer there → moving back to Step n if $m_n \geq 2$ → unloading a wafer from it → … → unloading the last wafer from Step n → moving to a loadlock → loading the wafer there⟩. Then, state W_{n+1} is reached and the close-down process is completed. Hence, the close-down process is defined as the process from W_1 to W_{n+1}. A PN model for the close-down process from state W_d to W_{d+1}, $d \in \mathbb{N}_n$, is shown in Fig. 6, where the meaning of the places and transitions is also identical to that in Fig. 4. Note that, in Fig. 6, place p_d modeling Step d is marked by k tokens with $1 \leq k \leq m_d$.

With the PN models in Figs. 4-6, we need to ensure that the backward strategy or generalized backward strategy can be realized with no deadlock. To do so, for the steady state, by following [30], a control policy is given as follows.

Definition 2: For the PN shown in Fig. 4, at any marking M, transition y_n is said to be control-enabled if $M(p_i) = m_i$, $i \in \mathbb{N}_n$; and y_i, $0 \leq i < n$, is said to be control-enabled if $M(p_{i+1}) = m_{i+1} - 1$.

With Definition 2, Wu *et al*. [30] show that the PN in Fig. 4 is deadlock-free under the steady state. Next, for the start-up process, a control policy is given as follows.

Definition 3: For the PN shown in Fig. 5, transition y_0 is said to be control-enabled if $M(p_1) < m_1$; and y_i, $1 \leq i < n$, is said to be control-enabled if $M(p_k) = m_k$, $k \in \mathbb{N}_i$, and $M(p_{i+1}) < m_{i+1}$.

By Definition 3, when a tool begins to work, $M(p_1) = 0 < m_1$, y_0 is control-enabled and it fires repeatedly until $M(p_1) = m_1$ such that State M_1 is reached. Then, y_1 is control-enabled due to that at this time $M(p_2) = 0 < m_2$ and $M(p_1) = m_1$. After firing y_1, $M(p_2) = 1$ and $M(p_1) = m_1 - 1$, y_0 is control-enabled again and it then fires. Repeat these processes until $M(p_2) = m_2$ and $M(p_1) = m_1$ such that State M_2 is reached. By continuing this process, finally State M_n is reached, that is to say that the generalized backward strategy for the start-up process can be realized with no deadlock.

For the close-down process, we also have a control policy which is given in Definition 4 below.

Definition 4: For the PN shown in Fig. 6, the following control policy is imposed:

1) y_n is said to be control-enabled if $M(p_i) = m_i$, $i \in \mathbb{N}_n \backslash \{1\}$;
2) y_i, $i \in \mathbb{N}_{n-1}$, is said to be control-enabled if $M(p_{i+1}) = m_{i+1} - 1$ and, $M(p_i) \neq 0$;
3) y_n is said to be control-enabled if $M(p_i) < m_i$, $i \in \mathbb{N}_{n-1} \backslash \{1\}$, $M(p_k) = 0$, $k \in \mathbb{N}_{i-1}$, and $M(p_j) = m_j$, $j \in \mathbb{N}_n \backslash \mathbb{N}_i$; and
4) y_n is said to be control-enabled if $M(p_n) > 0$, and $M(p_k) = 0$, $k \in \mathbb{N}_{n-1}$.

By Definition 4, when the close-down process begins, as $M(p_i) = m_i$, $i \in \mathbb{N}_n$, y_n is control-enabled. Firing y_n results in marking $M(p_n) = m_n - 1$ such that y_{n-1} is control-enabled. Firing y_{n-1} leads to $M(p_n) = m_n$ and $M(p_{n-1}) = m_{n-1} - 1$ such that y_{n-2} is control-enabled. In this way, y_{n-3} can fire and then y_{n-4}. Finally, y_1 is enabled and fires. Firing y_1 leads to $M(p_1) = m_1 - 1$ and

$M(p_i) = m_i$, $i \in \mathbb{N}_n\backslash\{1\}$, such that y_n is control-enabled again. By repeating the above processes, W_2 is then reached such that $M(p_1) = 0$ and $M(p_i) = m_i$, $i \in \mathbb{N}_n\backslash\{1\}$.

At State W_2, only y_n is control-enabled. Firing y_n transfers the system to a marking with $M(p_n) = m_n - 1$ such that y_{n-1} is control-enabled. Next, with y_{n-1}'s firing, y_{n-2} is control-enabled; then, y_{n-3}, until to y_2. By firing y_2, we reach marking $M(p_2) = m_2 - 1 < m_2$, $M(p_j) = m_j$, $j \in \mathbb{N}_n\backslash\mathbb{N}_2$, and $M(p_1) = 0$. Thus, by Step 3 of Definition 4, y_n is control-enabled. Repeat these processes until State W_3 is reached such that $M(p_1) = M(p_2) = 0$ and $M(p_i) = m_i$, $i \in \mathbb{N}_n\backslash\mathbb{N}_2$. In this way, we can reach State W_4 and finally State W_n.

At State W_n, as $M(p_k) = 0$, $k \in \mathbb{N}_{n-1}$ and $M(p_n) = m_n > 0$, y_n is control-enabled. After firing y_n, if $M(p_n) > 0$, y_n is still control-enabled. Then, y_n fires in a repeated way until $M(p_k) = 0$, $k \in \mathbb{N}_n$, or State W_{n+1} is reached. In this way, the generalized backward strategy for the close-down process can be realized with no deadlock.

D. Modeling Activity Time

To describe the temporal behavior of the PN models shown in Figs. 4-6, we associate both transitions and places with time. By following [3, 7], assume that:

1) The time taken by a robot loading/unloading activity at a step is a deterministic constant and denoted by λ; and
2) The time taken by a robot moving activity between two steps or two chambers at the same step with or without carrying a wafer is also a deterministic constant and denoted by μ.

Let α_i represent the wafer processing time at Step i, $i \in \mathbb{N}_n$. After being processed, a wafer can stay in a chamber for at most δ_i, $i \in \mathbb{N}_n$, time units. Let τ_i represent the wafer sojourn time at Step i, $i \in \mathbb{N}_n$, and it is required that $\tau_i \in [\alpha_i, \alpha_i+\delta_i]$ must hold. The time associated with different transitions and places is summarized in Table I.

Table I. Time duration associated with transitions and places.

Symbol	Transition or place	Action	Time duration
λ	$l_i \in T$	Loading a wafer into Step i, $i \in \mathbf{\Omega}_n$	λ
	$u_i \in T$	Unloading a wafer from Step i, $i \in \mathbf{\Omega}_n$	
μ	$z_i \in P$	Moving with a wafer being carried from Steps i - 1 to i, $i \in \mathbb{N}_{n+1}$	μ
μ	$y_i \in T$	Moving from a different step to Step i, $i \in \mathbf{\Omega}_n$, with no wafer being held, or moving between PCs at Step i, $i \in \mathbb{N}_n$, with no wafer being held	μ
τ_i	$p_i \in P$	A wafer stays in a chamber of p_i, $i \in \mathbb{N}_n$	$[\alpha_i, \alpha_i+\delta_i]$

With the WRTC, a feasible schedule is defined as follows.

Definition 5 [30]: In the PN models shown in Figs. 4-6, if the WRTC with time window $[\alpha_i, \alpha_i+\delta_i]$ is imposed on Step i, $i \in \mathbb{N}_n$, and by a schedule whenever u_i starts to fire, $\alpha_i \le \tau_i \le \alpha_i + \delta_i$ holds, then, this schedule is feasible.

III. Start-Up Process Scheduling

To test if a schedule is feasible, we need to calculate the wafer residency time in each chamber. At Step i, $i \in \mathbb{N}_n$, assume that the robot finishes loading a wafer into chamber at time instant ϕ_1 and the unloading of this wafer begins at time ϕ_2. Then, the wafer residency time at Step i is $\tau_i = \phi_2 - \phi_1$. Hence, for each wafer, to calculate τ_i, we need to determine ϕ_1 and ϕ_2. To do so, for the start-up process, let

1) $t_d(i, j, 2)$ denote the time when the robot starts to unload the jth wafer from Step i during the process from M_{d-1} to M_d, $d \in \mathbb{N}_n$, $i \in \mathbf{\Omega}_{n-1}$, and $1 \le j \le m_d$.
2) $t_d(i, j, 1)$ denote the time when the robot finishes loading the jth wafer into Step i during the process from M_{d-1} to M_d, $d \in \mathbb{N}_n$, $i \in \mathbb{N}_n$, and $1 \le j \le m_d$.
3) $\omega_d(i, j)$ denote the robot waiting time before it unloads the jth wafer from Step i during the process from M_{d-1} to M_d, $d \in \mathbb{N}_n$, $i \in \mathbf{\Omega}_{n-1}$, and $1 \le j \le m_d$.

Further, during the process from M_0 to M_1, let

$$t_1(0, 1, 2) = \omega_1(0, 1) = 0 \tag{3.1}$$

By (3.1), it means that the initial time is set to zero at which the robot starts to unload the first wafer from a loadlock. It follows from Table I that it takes λ time units to unload the wafer. Then, the robot moves this wafer to Step 1 and loads it there, which takes $\mu + \lambda$ time units. Hence, we have

$$t_1(1, 1, 1) = t_1(0, 1, 2) + 2\lambda + \mu \tag{3.2}$$

After the first wafer is loaded to Step 1, if $m_1 = 1$, State M_1 is reached; otherwise, if $m_1 > 1$, the robot needs to move back to a loadlock and unload another wafer. In this case, we have

$$t_1(0, j, 2) = t_1(1, j\text{-}1, 1) + \mu + \omega_1(0, j),\ 2 \le j \le m_1 \tag{3.3}$$

$$t_1(1, j, 1) = t_1(0, j, 2) + 2\lambda + \mu,\ 2 \le j \le m_1 \tag{3.4}$$

In this way, State M_1 is reached. Next, the model shown in Fig. 5 evolves from M_1 to M_2. During this process, if $m_1 = 1$, $t_2(1, 1, 2) = t_1(1, 1, 1) + \omega_2(1, 1)$; otherwise, $t_2(1, 1, 2) = t_1(1, m_1, 1) + \omega_2(1, 1) + \mu$, since that, with a "first-in first-out" rule, the robot moves from the chamber where the m_1th wafer is being processed to the chamber where the first wafer is being processed at Step 1, which takes μ time units. Also, if $m_1 = 1$ and $m_2 \ge 2$, during the process from States M_1 to M_2, after loading the jth wafer into Step 1 at time $t_2(1, j, 1)$, $1 \le j \le m_2$-1, the robot needs to wait at Step 1 until the wafer there is completed. Thus, we have

$$t_2(1, j, 2) = t_2(1, j\text{-}1, 1) + \omega_2(1, j),\ 2 \le j \le m_2$$

Similar to the process from States M_0 to M_1, with the generalized backward strategy, the system evolves from States M_{d-1} to M_d, $d \in \mathbb{N}_n\backslash\{1\}$, such that

$$t_d(d\text{-}1, 1, 2) = t_{d-1}(1, m_{d-1}, 1) + \mu + \omega_d(d\text{-}1, 1),\ 2 \le d \le n$$

and, if $m_1 = 1$,

$$t_2(1, 1, 2) = t_1(1, 1, 1) + \omega_2(1, 1) \tag{3.5}$$

$$t_d(d\text{-}1, j, 2) = t_d(1, j\text{-}1, 1) + \mu + \omega_d(d\text{-}1, j),\ 2 \le d \le n,\ 2 \le j \le m_d$$

and if $d = 2$ and $m_1 = 1$,

$$t_2(1, j, 2) = t_2(1, j\text{-}1, 1) + \omega_2(1, j),\ 2 \le j \le m_2 \tag{3.6}$$

$$t_d(d\text{-}m, j, 2) = t_d(d\text{-}m\text{+}2, j, 1) + \mu + \omega_d(d\text{-}m, j),\ 2 \le d \le n,\ 2 \le m \le d,\ 1 \le j \le m_d \tag{3.7}$$

$$t_d(d\text{-}m\text{+}1, j, 1) = t_d(d\text{-}m, j, 2) + 2\lambda + \mu,\ 2 \le d \le n,\ 1 \le m \le d,\ 1 \le j \le m_d \tag{3.8}$$

Expressions (3.5)-(3.7) present the time when the robot starts to unload a wafer from a step, while (3.8) gives the time when loading a wafer into a step ends. Note that in (3.6), if $d = 2$ and $m_1 = 1$, after loading a wafer into Step 1, the robot has to

wait there until the wafer finishes its processing, and then, unloads it, which means that no robot moving is required. Similarly, for (3.5), if $m_1 = 1$, no robot moving is required.

After loading the m_nth wafer into Step 1 at time $t_n(1, m_n, 1)$, there is a wafer in processing in each chamber such that State M_n (the steady state) is reached. With $t_n(1, m_n, 1)$ as the starting time of the steady state, we use c_1 to denote the first steady cycle that ends at the time when a wafer has just loaded into Step 1 during the steady state process. With c_d denoting the dth steady cycle, the system then goes into steady cycle c_2. For the following presentation, let

1) $t_{n+d}(i, 1, 2)$ denote the time when the robot starts to unload a wafer from Step i during steady cycle c_d, $d \geq 1$.
2) $t_{n+d}(i, 1, 1)$ denote the time when the robot finishes loading a wafer into Step i during steady cycle c_d, $d \geq 1$, and
3) $\omega_{n+d}(i, 1)$ denote the robot waiting time before unloading a wafer from Step i during steady cycle c_d, $d \geq 1$.

Note that Step 0 is same as Step n+1. During the start-up process, to complete the process as soon as possible, the robot waiting time at Step i, $i \in \mathbf{\Omega}_n$, needs to schedule to be different. However, under the steady state, they should be identical for different cycles. To ensure that the system can operate in a periodic way in the steady state, we let $\mathfrak{R} = \max\{m_i | i \in \mathbb{N}_n\}+1$. Then, similar to (3.5)-(3.8) we have

$$t_{n+1}(n, 1, 2) = t_n(1, m_n, 1) + \mu + \omega_{n+1}(n, 1) \quad (3.9)$$

$$t_{n+j}(n, 1, 2) = t_{n+j-1}(1, 1, 1) + \mu + \omega_{n+j}(n, 1), 2 \leq j \leq \mathfrak{R} \quad (3.10)$$

$$t_{n+j}(0, 1, 1) = t_{n+j}(n, 1, 2) + 2\lambda + \mu, 1 \leq j \leq \mathfrak{R} \quad (3.11)$$

$$t_{n+j}(n-m, 1, 2) = t_{n+j}(n-m+2, 1, 1) + \mu + \omega_{n+j}(n-m, 1), \quad 1 \leq j \leq \mathfrak{R}, 1 \leq m \leq n \quad (3.12)$$

$$t_{n+j}(n-m+1, 1, 1) = t_{n+j}(n-m, 1, 2) + 2\lambda + \mu, \quad 1 \leq j \leq \mathfrak{R}, 1 \leq m \leq n \quad (3.13)$$

To ensure that the waiting time for different cycles under the steady state is identical, we have

$$\omega_{n+j}(m, 1) = \omega_{n+k}(m, 1), m \in \mathbf{\Omega}_n, j \neq k \text{ and } j, k \in \mathbb{N}_{\mathfrak{R}} \quad (3.14)$$

Thus, by Constraints (3.9)-(3.14), it is sure that the obtained schedule for the start-up process is compatible to a cyclic schedule under the steady state. Since the robot waiting time should be nonnegative, we have

$$\omega_{n+j}(m, 1) \geq 0, j \in \mathbb{N}_{\mathfrak{R}} \quad (3.15)$$

$$\omega_d(d-m, j) \geq 0, 1 \leq d \leq n, 1 \leq m \leq d, 1 \leq j \leq m_d \quad (3.16)$$

It follows from the above analysis that, during the start-up process, the robot waiting time at Step i, $i \in \mathbf{\Omega}_n$, is varied when different wafers processed by different chambers at the same step are unloaded. Thus, the wafer residency time in different chambers at the same step is different, which makes the scheduling problem of transient process of single-arm cluster tool with parallel chambers more challenging and complicated than a tool with single PC for a process step. Also, to ensure that a robot task sequence during the transient process should be always compatible with the given steady-state cyclic schedule is another challenging issue. In this work, the generalized backward strategy becomes a backward strategy that adopted by the steady-state cyclic schedule, when there is a wafer in each PC. Furthermore, by (3.9)-(3.14), compatibility issue is ensured too.

In order to calculate the wafer sojourn time in a chamber, we assume that a wafer is loaded into Step i by executing $t_d(i, j, 1)$ with d, i and j being deterministic. With $M(p_k)$ representing the number of tokens (wafers) in p_k after $t_d(i, j, 1)$ is performed, if $\sum_{k=i}^{n} M(p_k) \leq \sum_{l=i+1}^{n} m_l$, we assume that the wafer is unloaded by performing $t_q(i, z, 2)$. Then, the values of q and z can be calculated as follows:

1) Find the value of q such that $\sum_{l=i+1}^{q-1} m_l < \sum_{k=i}^{n} M(p_k) \leq \sum_{l=i+1}^{q} m_l$;
2) $z = \sum_{k=i}^{n} M(p_k) - \sum_{l=i+1}^{q-1} m_l$.

With $\sum_{l=i+1}^{q-1} m_l < \sum_{k=i}^{n} M(p_k) \leq \sum_{l=i+1}^{q} m_l$, we can conclude that the $(\sum_{k=i}^{n} M(p_k) - \sum_{l=i+1}^{q-1} m_l)$th wafer in a PC at Step i is unloaded during the state evolution process from M_{q-1} to M_q.

With q and z being determined, to meet the WRTC, we have the following constraint.

$$\alpha_i \leq t_q(i, z, 2) - t_d(i, j, 1) \leq \alpha_i + \delta_i \quad (3.17)$$

However, if $\sum_{k=i}^{n} M(p_k) > \sum_{l=i+1}^{n} m_l$, we have that this wafer has been unloaded, which was performed by starting at time $t_{n+r}(i, 1, 2)$, where $r = \sum_{k=i}^{n} M(p_k) - \sum_{l=i+1}^{n} m_l$. Hence, we have the following constraint.

$$\alpha_i \leq t_{n+r}(i, 1, 2) - t_d(i, j, 1) \leq \alpha_i + \delta_i, 1 \leq r \leq \mathfrak{R} \quad (3.18)$$

Then, we present the following Linear Program denoted as LP1 to schedule the start-up process.

$$\text{Min}\{\sum_{j=1}^{m_d}\sum_{m=1}^{d}\sum_{d=1}^{n}\omega_d(d-m, j) + \sum_{j=1}^{\mathfrak{R}}\sum_{m=0}^{n}\omega_{n+j}(m,1)\} \quad (3.19)$$

Subject to: (3.1)-(3.18).

In LP1, the objective is to minimize the total robot waiting time. Generally, a linear program can be efficiently solved. Moreover, in practice, a cluster tool usually consists of no more than six PCs such that size of the above linear program relevantly small. Thus, it can be very efficiently solved by using a commercial solver, i.e., it is computationally efficient.

During the start-up process, with a generalized backward strategy, both the robot task sequence and activity (including unloading, moving and loading) time are deterministic and known. What we need to do for scheduling such a process is to determine the robot waiting time at the steps. The sum of time taken for the robot actions and waiting is the time taken for start-up process. By (3.19), the total robot waiting time is minimized. Thus, by solving LP1, the makespan for the start-up process is minimized. Furthermore, the WRTC is embedded into (3.17) and (3.18), i.e., the feasibility for the obtained schedule is ensured. Hence, we have the following result immediately.

Property 3.1: For the start-up process of a single-arm cluster tool with WRTC, if the generalized backward strategy is applied and a solution is found by solving LP1, then the obtained schedule is optimal and feasible.

With Property 3.1, the remaining question is if an optimal and feasible solution can be always found by solving LP1 for a single-arm tool. To answer this question, we introduce the virtual wafer method proposed by Wu *et al.* [30] for the

start-up and close-down processes. In [30], the virtual wafer method is proposed to implement an obtained cyclic schedule under the steady state for cluster tools with WRTC.

It should be pointed out that, at most of the time, a cluster tool operates under a steady state. Thus, after the start-up process ends, a tool enters a steady state and operates in a periodical way, also a close-down process begins when a steady state process ends. Hence, it is meaningful to schedule the start-up and close-down processes only a cyclic schedule under a steady state exists.

During the start-up process, with the virtual wafer method, it is assumed that there is a virtual wafer in each chamber at the beginning such that, during the process from the idle state to the steady state, a tool operates exactly according to a cyclic schedule under the steady state (such a schedule can be obtained using the method proposed by Wu *et al*. [30]). It assumed that initially there is a virtual wafer in each PC, during the start-up process, each time when the robot needs to unload a wafer from a loadlock, it unloads a real one and when it requires to load a wafer into a loadlock, it loads a virtual one. After all virtual wafers are removed from the system, it reaches the steady state. Note that, in this work, it is assumed that there is a feasible cyclic schedule for a time-constrained single-arm cluster tool. This assumption is reasonable since we do not need to optimize the start-up process if there is no feasible cyclic schedule. Then, we have the following result.

Theorem 3.1: For a single-arm cluster tool with WRTC, an optimal and feasible schedule for the start-up process can be obtained by solving LP1 if, under the steady state, a feasible periodic schedule exists.

Proof: For a single-arm cluster tool under the steady state, as the wafer processing time is much longer than the robot activity time, a backward strategy is optimal. With such a strategy, if there is a feasible periodic schedule, then, by using a virtual wafer method, during the process from state M_0 to M_1, operations $B_1(1, 1)$, $B_1(1, 2)$, …, and $B_1(1, m_1)$ are performed sequentially. According to the backward strategy and the PN model in Fig. 4, after performing $B_1(1, 1)$, the robot task sequence $\langle y_n \to u_n \to z_n \to l_0 \to y_{n-1} \to u_{n-1} \to z_{n-1} \to l_n \to y_{n-2} \to \ldots \to y_1 \to u_1 \to z_1 \to l_2 \to y_0 \to B_1(1, 2)\rangle$ is executed. In this sequence, tasks from y_n to l_2 are used to process virtual wafers, i.e., they are virtual operations. The time taken to perform these virtual operations can be seen as waiting time at Step 1. After performing this sequence, the robot moves to a loadlock by firing y_0 and it is followed by $B_1(1, 2)$.

Similarly, during the process from state M_0 to M_1, the time taken for performing the virtual operations can be seen as waiting time at Step 1. In this way, state M_1 can be reached and the WRTC must be satisfied.

Next, from state M_d to M_{d+1}, $2 \le d \le n - 1$, we can also treat the time taken for performing virtual operations as waiting time at Step d and the WRTC should always be met. The start-up process ends when M_n is reached.

From the above analysis we know that the obtained schedule by the method given in [30] is in the feasible region of LP1. Therefore, the theorem holds. ■

IV. Close-Down Process Scheduling

For the close-down process scheduling, we let

1) $T_d(i, j, 2)$ denote the time when the robot starts to unload the jth wafer from a PC at Step i during the process from W_d to W_{d+1}, $d \in \mathbb{N}_n$, $i \in \mathbb{N}_n$, and $1 \le j \le m_d$;
2) $T_d(i, j, 1)$ denote the time when the robot finishes loading the jth wafer into a PC at Step i during the process from W_d to W_{d+1}, $d \in \mathbb{N}_n$, $i \in \mathbb{N}_{n+1}\backslash\{1\}$, and $1 \le j \le m_d$; and
3) $w_d(i, j)$ denote the robot waiting time before unloading the jth wafer from a PC at Step i during the process from W_d to W_{d+1}, $d \in \mathbb{N}_n$, $i \in \mathbb{N}_n$, and $1 \le j \le m_d$.

Let T_0 (= 0) denote the time instant when the last wafer is loaded into Step 1 during the operation under the steady state, i.e., the time when the steady state operation ends and the close-down process starts. Then, with a generalized backward strategy stated in Section II, during the process from States W_1 to W_2, the robot goes to Step n with μ time units being taken and waits there for $w_1(n, 1)$ time units, i.e., we have

$$T_1(n, 1, 2) = T_0 + \mu + w_1(n, 1) = \mu + w_1(n, 1) \qquad (4.1)$$

After executing $T_1(n, 1, 2)$, we can easily obtain

$$T_1(0, 1, 1) = T_1(n, 1, 2) + 2\lambda + \mu,$$
$$T_1(n\text{-}1, 1, 2) = T_1(0, 1, 1) + \mu + w_1(n\text{-}1, 1),$$
$$T_1(n, 1, 1) = T_1(n\text{-}1, 1, 2) + 2\lambda + \mu, \ldots, \text{and},$$
$$T_1(2, 1, 1) = T_1(1, 1, 2) + 2\lambda + \mu.$$

Thereafter, if $m_1 \ge 2$, the robot goes to Step n to perform $T_1(n, 2, 2)$, or we have

$$T_1(n, 2, 2) = T_1(2, 1, 1) + \mu + w_1(n, 2).$$

In this way, after dropping the m_1th wafer into Step 2 at time $T_1(2, m_1, 1) = T_1(1, m_1, 2) + 2\lambda + \mu$, State W_2 is reached. Next, the system evolves from States W_2 to W_3 and the robot goes to Step n for unloading the 1st wafer from Step n at time $T_2(n, 1, 2)$. Thus, we have

$$T_2(n, 1, 2) = T_1(2, m_1, 1) + \mu + w_2(n, 1)$$

For the general case, we have

$$T_d(n, 1, 2) = T_{d\text{-}1}(d, m_{d\text{-}1}, 1) + \mu + w_d(n, 1), \quad 2 \le d \le n\text{-}1 \qquad (4.2)$$
$$T_d(n, j, 2) = T_d(d+1, j\text{-}1, 1) + \mu + w_d(n, j), \quad 1 \le d \le n\text{-}1,\ 2 \le j \le m_d$$

and if, $d = n$ -1, and $m_n = 1$,

$$T_{n\text{-}1}(n, j, 2) = T_{n\text{-}1}(n, j\text{-}1, 1) + w_{n\text{-}1}(n, j),\ 2 \le j \le m_{n\text{-}1} \qquad (4.3)$$
$$T_d(0, j, 1) = T_d(n, j, 2) + 2\lambda + \mu,\ 1 \le d \le n\text{-}1,\ 1 \le j \le m_d \qquad (4.4)$$
$$T_d(n\text{-}m, j, 2) = T_d(n\text{-}m+2, j, 1) + \mu + w_d(n\text{-}m, j), \quad 1 \le d \le n\text{-}1,\ 1 \le m \le n\text{-}d,\ 1 \le j \le m_d \qquad (4.5)$$
$$T_d(n\text{-}m+1, j, 1) = T_d(n\text{-}m, j, 2) + 2\lambda + \mu,\ 1 \le d \le n\text{-}1, \quad 1 \le m \le n\text{-}d,\ 1 \le j \le m_d \qquad (4.6)$$

Note that in (4.3), if $d = n$ -1 and $m_n = 1$, after loading a wafer into Step n, the robot needs to wait there until this wafer completes its processing and then unloads it, which means that no robot moving is required.

After loading the m_{n-1}th wafer into Step n at time $T_{n-1}(n, m_{n-1}, 1)$, State W_n is reached. Then, the system evolves from States W_n to W_{n+1}, at this time all PCs are emptied, i.e., the closed-down process ends. During this process, we have

$$T_n(n, 1, 2) = T_{n\text{-}1}(n, m_{n\text{-}1}, 1) + w_n(n, 1)$$

or

$$T_n(n, 1, 2) = T_{n\text{-}1}(n, m_{n\text{-}1}, 1) + w_n(n, 1) + \mu, \text{ if } 2 \le m_n \qquad (4.7)$$

At time $T_n(n, 1, 2)$, a wafer has been unloaded from a PC at Step n and is moved and loaded to a loadlock, or we have

$$T_n(0, 1, 1) = T_n(n, 1, 2) + 2\lambda + \mu$$

Next, if $m_n \geq 2$, the robot goes back to Step n again and unloads another wafer from it, or we have

$$T_n(n, 2, 2) = T_n(0, 1, 1) + w_n(n, 2) + \mu$$

For a general case, we have:

$$T_n(n, j, 2) = T_n(0, j\text{-}1, 1) + w_n(n, j) + \mu,\ 2 \leq j \leq m_n \quad (4.8)$$

and $$T_n(0, j, 1) = T_n(n, j, 2) + 2\lambda + \mu,\ 1 \leq j \leq m_n \quad (4.9)$$

To make sure that the waiting time at a step is nonnegative, we have (4.10) and (4.11).

$$w_d(d\text{-}m, j) \geq 0,\ 1 \leq d \leq n\text{-}1,\ 0 \leq m \leq n\text{-}d,\ 1 \leq j \leq m_d \quad (4.10)$$

$$w_n(n, j) \geq 0,\ 1 \leq j \leq m_n \quad (4.11)$$

With T_0 (= 0) being the time when the close-down process starts, there are two different cases for calculating the wafer sojourn time at a chamber:

Case 1: Wafers are loaded into a chamber before T_0, and

Case 2: Wafers are loaded into a chamber after T_0.

For Case 1, we need to know how long the wafers have already stayed in the chambers. To do so, let w_i, $i \in \mathbf{\Omega}_n$, denote the robot waiting time at Step i in the steady state, whose value can be determined by LP1, or $w_i = \omega_{n+j}(m, 1)$, $m \in \mathbf{\Omega}_n$, and $j \in \mathbb{N}_{\mathfrak{R}}$. Further, let ψ denote the system cycle time under the steady state. Then, according to Wu *et al*. [30], we have

$$\psi = 2(\lambda + \mu) + \sum_{i=0}^{n} w_i \quad (4.12)$$

Assume that, for Step i, $i \in \mathbb{N}_n$, Chamber 1 is occupied by the first wafer, Chamber 2 the second, … , and Chamber m_i the last. At T_0, let Υ_{ij}, $i \in \mathbb{N}_n$ and $1 \leq j \leq m_i$, denote the time duration for which a wafer has already stayed in Chamber j at Step i. For Chamber m_1 at Step 1, due to that the robot finishes its loading at T_0, we have $\Upsilon_{1(m_1)} = 0$. For a general case, we have

$$\Upsilon_{1j} = (m_1 - j) \times \psi,\ 1 \leq j \leq m_1 \quad (4.13)$$

To calculate $\Upsilon_{2(m_2)}$, we have the fact that after loading the last wafer into m_2, the robot executes task sequence ⟨moving to the loadlocks → waiting with time w_0 → unloading a wafer from it → moving to Step 1 → loading that wafer there⟩ which takes $2(\lambda + \mu) + w_0$ time units. Thus, we have

$$\Upsilon_{2(m_2)} = 2(\lambda + \mu) + w_0$$

Similar to (4.13), we have

$$\Upsilon_{2j} = 2(\lambda + \mu) + w_0 + (m_2 - j) \times \psi,\ 1 \leq j \leq m_2$$

For a general case, we have

$$\Upsilon_{ij} = 2 \times (i - 1) \times (\lambda + \mu) + \sum_{k=0}^{i-2} w_k + (m_i - j) \times \psi,\ 2 \leq i \leq n,\ 1 \leq j \leq m_i \quad (4.14)$$

With (4.13) and (4.14), we can calculate the wafer sojourn time as follows. For a wafer that is being processed in Chamber k at Step i, $i \in \mathbb{N}_n$ and $1 \leq k \leq m_i$, with k and i being deterministic, we first find the value of o such that $k \in (\sum_{p=1}^{o-1} m_p, \sum_{p=1}^{o} m_p]$. This wafer is unloaded during the state evolution process from W_o to W_{o+1}.

If $\|[1, m_i] \cap (\sum_{p=1}^{o-1} m_p, \sum_{p=1}^{o} m_p]\| = 1$, or $[1, m_i] \cap (\sum_{p=1}^{o-1} m_p, \sum_{p=1}^{o} m_p] = k$, then this wafer is unloaded from Step i, which is performed by starting at time $T_o(i, 1, 2)$. Hence, we have the following constraint.

$$\alpha_i \leq T_o(i, 1, 2) + \Upsilon_{ik} \leq \alpha_i + \delta_i,\ 1 \leq i \leq n,\ 1 \leq j \leq m_i \quad (4.15)$$

If $\|[1, m_i] \cap (\sum_{p=1}^{o-1} m_p, \sum_{p=1}^{o} m_p]\| > 1$, we first let $[1, m_i] \cap (\sum_{p=1}^{o-1} m_p, \sum_{p=1}^{o} m_p] = \{k_1, k_1+1, k_1+2, \ldots, k_1+j\}$, where the value of k_1 is given and the wafer in Chamber k_1 is loaded earlier than the one in Chamber k_1+1, and the one in k_1+p is loaded earlier than the one in k_1+p+1, $1 \leq p \leq j\text{-}1$. Then, we replace the numbers k_1, k_1+1, k_1+2, …, and k_1+j by 1, 2, …, and j+1. In this way, to satisfy the WRTC, we have the following constraint.

$$\alpha_i \leq T_o(i, h, 2) + \Upsilon_{i(k_1+h\text{-}1)} \leq \alpha_i + \delta_i,\ h \in \mathbb{N}_{j+1} \quad (4.16)$$

For Case 2, assume that a jth wafer has just loaded into Step i at time $T_d(i, j, 1)$ with i, j, and d being known, and this wafer is unloaded at time $T_o(i, h, 2)$, where i is known and, o and h need to be determined.

To determine the value of o and h, we have the following results:

1) If there is an s such that $\sum_{r=1}^{s} m_{i-r} < \sum_{k=1}^{i-1} M(p_k) - (m_i - 1) \leq \sum_{r=1}^{s+1} m_{i-r}$, then $o = i - (s + 1)$ and $h = m_i + m_o - \sum_{k=1}^{o} M(p_k) + 1$;
2) If $0 < \sum_{k=1}^{i-1} M(p_k) - (m_i - 1) \leq m_{i\text{-}1}$, then $o = i - 1$ and $h = m_i + m_o - \sum_{k=1}^{o} M(p_k)$; and
3) If $\sum_{k=1}^{i-1} M(p_k) - (m_i - 1) \leq 0$, then $o = i$ and $h = m_i - \sum_{k=1}^{i-1} M(p_k)$.

where $M(p_k)$ represents the number of tokens (wafers) in p_k at time $T_d(i, j, 1)$. By doing so, to meet the WRTC, we have Constraint (4.17).

$$\alpha_i \leq T_o(i, h, 2) - T_d(i, j, 1) \leq \alpha_i + \delta_i \quad (4.17)$$

Then, we have Linear Program 2 (LP2) for scheduling the close-down process as follows.

$$\text{Min}\{\sum_{j=1}^{m_d}\sum_{m=0}^{n-d}\sum_{d=1}^{n-1} w_d(n-m, j) + \sum_{j=1}^{m_n} w_n(n, j)\} \quad (4.18)$$

Subject to: (4.1)-(4.11) and (4.15)-(4.17).

Similar to LP1, LP2 is computationally efficient and we also have the following results for it.

Property 4.1: For the close-down process of a single-arm cluster tool with WRTC, by using the generalized backward strategy, a solution obtained by solving LP2 is optimal and feasible.

Theorem 4.1: For a single-arm cluster tool with WRTC, an optimal and feasible schedule for the close-down process can be obtained by solving LP2 if there exists a feasible cyclic schedule under the steady state.

Proof: By treating the time taken for performing virtual operations as waiting time, it can be shown similarly to that

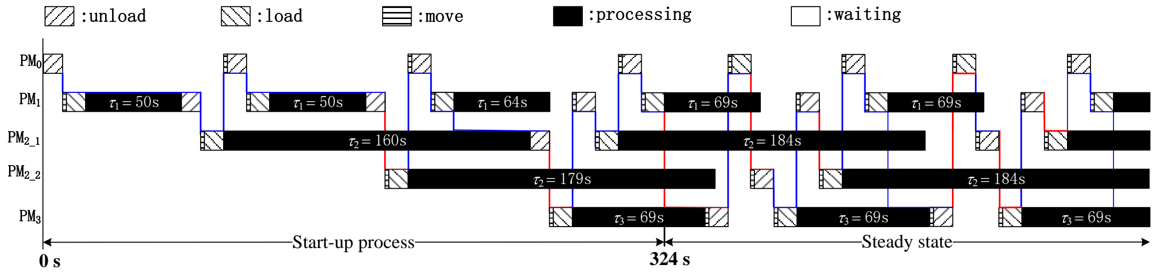


Fig. 7. Simulation result for Example 1.

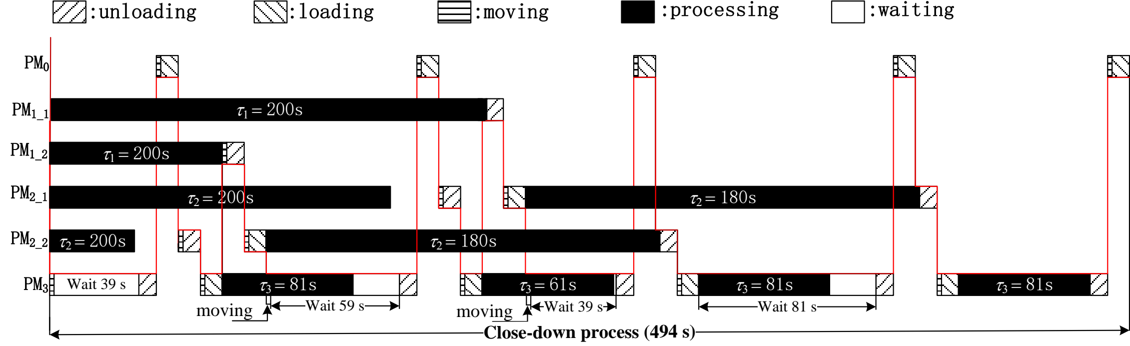


Fig. 8. Simulation result for Example 2.

for Theorem 3.1. ■

V. Illustrative Examples

Example 1: A cluster tool has a wafer flow pattern (1, 2, 1). It takes 50, 160, and 69 time units to process a wafer in a PC at Steps 1, 2, and 3, respectively. After being processed, a wafer can stay in the PC at Steps 1, 2, and 3 for at most 20, 26 and 15 time units, respectively. It needs 10 time units for the robot to unload/load a wafer from/into a PC and 2 time units to move between two different chambers with or without holding a wafer, i.e.,

$\alpha_1 = 50$, $\alpha_2 = 160$, $\alpha_3 = 69$;

$\delta_1 = 20$, $\delta_2 = 26$, $\delta_3 = 15$; and

$\lambda = 10$, and $\mu = 2$.

For this example, by Wu *et al.* [30], a feasible cyclic schedule under the steady state with cycle time 115 is obtained by setting the robot waiting as $w_0 = w_1 = w_2 = 0$, and $w_3 = 19$. With the virtual wafer method proposed by Wu *et al.* [30], it takes 367 time units to finish the start-up process.

With LP1, an optimal and feasible schedule for the start-up process is obtained with robot waiting time being set as shown in Table II. Thereafter, the system enters its steady state and the robot waiting time is set as

$\omega_4(3, 1) = \omega_5(3, 1) = \omega_6(3, 1) = 19$,

$\omega_4(2, 1) = \omega_5(2, 1) = \omega_6(2, 1) = 0$,

$\omega_4(1, 1) = \omega_5(1, 1) = \omega_6(1, 1) = 0$, and,

$\omega_4(0, 1) = \omega_5(0, 1) = \omega_6(0, 1) = 0$.

Note that they are exactly identical to that for the schedule under the steady state, i.e., this start-up process is compatible with the cyclic schedule. Its simulation result is shown in Fig. 7 and it takes 324 time units to complete the start-up process. Compared with the result obtained by Wu *et al.* [30], it shortens the time by 11.7% and it is significant.

Table II. Solution for the start-up process of Example 1.

Start-up process	**Robot waiting time setting**
From States M_0 to M_1	$\omega_1(0, 1) = 0$
From States M_1 to M_2	$\omega_2(1, 1) = 50$, $\omega_2(0, 1) = 0$, $\omega_2(1, 2) = 50$, $\omega_2(0, 2) = 0$
From States M_2 to M_3	$\omega_3(2, 1) = 38$, $\omega_3(1, 1) = 0$, $\omega_3(0, 1) = 0$

Example 2: A cluster tool has a wafer flow pattern (2, 2, 1). It takes 200, 180, and 86 time units to process a wafer in a PC at Steps 1, 2, and 3, respectively. After being processed, a wafer can stay in the PC at Steps 1, 2, and 3 for at most 18, 20 and 22 time units, respectively. It needs 8 time units for the robot to unload/load a wafer from/into a PC and 2 time units to move between two different chambers with or without holding a wafer, i.e.,

$\alpha_1 = 200$, $\alpha_2 = 180$, $\alpha_3 = 60$;

$\delta_1 = 18$, $\delta_2 = 20$, $\delta_3 = 22$; and

$\lambda = 8$, and $\mu = 2$.

For this example, by Wu *et al.* [30], a feasible cyclic schedule under the steady state with cycle time 119 is obtained by setting the robot waiting time as $w_0 = w_1 = w_2 = 0$, and $w_3 = 39$. With the virtual wafer method proposed by Wu *et al.* [30], it takes 535 time units to complete the close-down process. Note that the close-down process starts at the time when the last wafer is loaded into Step 1.

Table III. Solution for close-down process of example 2.

Close-down process	**Robot waiting time setting**
From States W_1 to W_2	$w_1(3, 1) = 39$, $w_1(2, 1) = 0$, $w_1(1, 1) = 0$; $w_1(3, 2) = 59$, $w_1(2, 2) = 0$, $w_1(1, 2) = 0$
From States W_2 to W_3	$w_2(3, 1) = 39$, $w_2(2, 1) = 0$; $w_2(3, 2) = 81$, $w_2(2, 2) = 0$
From States W_3 to W_4	$w_3(3, 1) = 60$

With LP2, an optimal and feasible schedule for the close-down process is obtained with robot waiting time being set as shown in Table III. Note that the robot waiting time setting is compatible with that for the steady state schedule.

Its simulation result is shown in Fig. 8 and it takes 494 time

units to complete the closed-down process. Compared with the result obtained by Wu *et al*. [30], it is reduced by 7.7%.

VI. Conclusions

Cluster tools are widely used to process wafers in semiconductor manufacturing. Recently, due to diversified customer demands, as well as corrective and preventive maintenance, there are frequent close-down and start-up processes for cluster tools. Some wafer manufacturing processes which are characterized by wafer residency time constraint require that a processed wafer should be removed from a chamber in a limited time. Such a requirement makes the scheduling problem of cluster tools more challenging. This work represents the first one that studies the challenging problem of scheduling both start-up and close-down processes for time-constrained single-arm cluster tools with parallel chambers for a processing step. Based on a generalized backward strategy, this work first develops two timed Petri net models to reveal the dynamic behavior during the two transient processes. Based on these models, two linear programs are formulated to search an optimal and feasible schedule. As a result, an optimal and feasible schedule for both processes can be efficiently found.

In this work, all the activity time is seen as deterministic. However, they may be subject to random variation, leading to residency time fluctuation in a chamber, which could make a feasible schedule obtained under fixed activity time infeasible. Hence, the scheduling problem becomes more complex and challenging and it is our future work. It is also very meaningful to extend the obtained result to dual-arm cluster tools.

References

[1] L. P. Bai, N. Q. Wu, Z. W. Li, and M. C. Zhou, "Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology," *IEEE Transactions on Systems*, *Man*, *& Cybernetics*: *Systems*, vol. 46, no. 10, 1456-1467, Oct. 2016.

[2] L. Ferrarini and L. Piroddi, "Modeling and control of fluid transportation operations in production plants with Petri nets," *IEEE Trans. Control Syst. Technol*., vol. 16, no. 5, pp. 1090-1098, Sep. 2008.

[3] H. N. Geismar, C. Sriskandarajah, and N. Ramanan, "Increasing throughput for robotic cells with parallel machines and multiple robots," *IEEE Transaction on Automation Science and Engineering*, vol. 1, no. 1, 84-89, 2004.

[4] Y. Hou, N. Q. Wu, M. C. Zhou, and Z. W. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Transactions on Systems*, *Man*, *& Cybernetics*: *Systems*, vol. 47, no. 3, 517-530, Mar. 2017.

[5] H. Hu and M. C. Zhou, "A Petri Net-based Discrete Event Control of Automated Manufacturing Systems with Assembly Operations," *IEEE Trans. on Control Systems Technology*, vol. 23, no. 2, 513-524, Mar. 2015.

[6] H. Hu, Y. Liu and M. C. Zhou, "Maximally Permissive Distributed Control of Large Scale Automated Manufacturing Systems Modeled with Petri Nets," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 2026-2034, Sept. 2015.

[7] J.-H. Kim, T.-E. Lee, H.-Y. Lee, and D.-B. Park, "Scheduling analysis of timed-constrained dual-armed cluster tools," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 3, 521-534, 2003.

[8] J.-H. Kim, M. C. Zhou, and T. E. Lee, "Schedule Restoration for Single-Armed Cluster Tools," *IEEE Trans. on Semiconductor Manufacturing*, vol.27, no. 3, pp. 388-399, Aug. 2014.

[9] H.-J. Kim, J.-H. Lee, and T.-E. Lee, "Noncyclic scheduling of cluster tools with a branch and bound algorithm," *IEEE Trans. Autom. Sci. Eng*., vol. 12, no. 2, pp. 690–700, Apr. 2015.

[10] H.-J. Kim, J.-H. Lee, and T.-E. Lee, "Time-feasible reachability tree for noncyclic scheduling of timed petri nets," *IEEE Trans. Autom. Sci. Eng., IEEE Trans. Autom. Sci. Eng*., vol. 12, no. 3, pp. 1007-1016, Jul. 2015.

[11] D.-K. Kim, T.-E. Lee, and H.-J. Kim, "Optimal scheduling of transient cycles for single-armed cluster tools with parallel chambers," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 1165-1175, 2016.

[12] J. H. Lee, H. J. Kim, and T. E. Lee, "Scheduling lot switching operations for cluster tools," *IEEE Trans. Semicond. Manuf*., vol. 26, no. 4, pp. 592–601, Apr. 2013.

[13] J. H. Lee and H. J. Kim, "Makespan analysis of lot switching period in cluster tools," *IEEE Trans. Semicond. Manuf*., vol. 29, no. 2, pp. 127-136, May 2016.

[14] J. H. Lee and H. J. Kim, "Completion time analysis of wafer lots in single-armed cluster tools with parallel processing modules," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1622-1633, Oct. 2017.

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

[16] H. W. Liao, Y. Wang, J. Stanley, S. Lafortune, S. Reveliotis, T. Kelly, S. Mahlke, "Eliminating concurrency bugs in multithreaded software: A new approach based on discrete-event control," *IEEE Trans. Control Syst. Technol*., vol. 21, no. 6, pp. 2067–2082, Nov. 2013.

[17] M.-J. Lopez and S.-C. Wood, "Systems of multiple cluster tools - configuration, reliability, and performance," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 2, 170-178, 2003.

[18] X. Y. Lu, M. C. Zhou, A. C. Ammari, and J. C. Ji, "Hybrid Petri nets for modeling and analysis of microgrid systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 4, pp. 347-354, Oct. 2016.

[19] J. Luo and M. C. Zhou, "Petri-net controller synthesis for partially controllable and observable discrete event systems," *IEEE Trans. on Automatic Control*, vol. 62, no. 3, pp. 1301 – 1313, Mar. 2017.

[20] T. Nishi, and R. Maeno, "Petri net decomposition approach to optimization of route planning problems for AGV systems," *IEEE Trans. Autom. Sci. Eng*., vol. 7, no. 3, pp. 523-537, Jul. 2010.

[21] T. Nishi and Y. Tanaka, "Petri net decomposition approach for dispatching and conflict-free routing of bidirectional automated guided vehicle systems," *IEEE Transactions on Systems, Man, and Cybernetics- Part A: Systems and Humans*, vol. 42, no. 5, pp. 1230-1243, Sep. 2012.

[22] T. Nishi, and I. Matsumoto, "Petri net decomposition approach to deadlock-free and non-cyclic scheduling of dual-armed cluster tools," *IEEE Trans. Autom. Sci. Eng*., vol. 12, no. 1, pp. 281-294, Jan. 2015.

[23] C. R. Pan, M. C. Zhou, Y. Qiao, and N. Q. Wu, "Scheduling cluster tools in semiconductor manufacturing: recent advances and challenges," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, 586-601, Apr. 2018.

[24] L. Qi, M. C. Zhou, and W. J. Luan, "Emergency traffic-light control system design for intersections subject to accidents," *IEEE Trans. on Intelligent Transportation Systems,* vol. 17, no. 1, pp. 170-183, Jan. 2016.

[25] Y. Qiao, C. R. Pan, N. Q. Wu, and M. Zhou, "Response policies to process module failure in single-arm cluster tools subject to wafer residency time constraints," *IEEE Trans. Autom. Sci. Eng*., vol. 12, no. 3, pp. 1125–1139, Jul. 2015.

[26] Y. Qiao, M. C. Zhou, N. Q. Wu, and Q. H. Zhu, "Scheduling and control of startup process for single-arm cluster tools with residency time constraints," *IEEE Trans. Control Syst. Technol*., vol. 25, no. 4, pp. 1243-1256, Jul., 2017.

[27] N. Ran, H. Y. Su, S. G. Wang, "An improved approach to test diagnosability of bounded Petri nets," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 2, 297-303, 2017.

[28] 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.

[29] N. Q. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Transaction on Systems*, *Man*, *and Cybernetics*, *Part C*, vol. 29, no. 2, pp. 192-204, 1999.

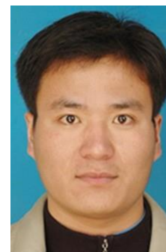
[30] N. Q. Wu, C. B. Chu, F. Chu, and M. C. Zhou, "A Petri net method for schedulability and scheduling problems in single-arm cluster tools with

wafer residency time constraints," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 2, pp. 224-237, 2008.
[31] N. Q. Wu, F. Chu, C. Chu, and M. Zhou, "Petri net-based scheduling of single-arm cluster tools with reentrant atomic layer deposition processes," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 42-55, 2011.
[32] 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 Transactions on Systems*, *Man*, *& Cybernetics*: *Systems*, vol. 43, no. 1, 196-207, Jan. 2013.
[33] N. Q. Wu, Z. W. Li, and T. Qu, "Energy efficiency optimization in scheduling crude oil operations of refinery based on linear programming," *Journal of Cleaner Production*, vol. 166, 49-57, Nov. 2017.
[34] N. Q. Wu and M. C. Zhou, "Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems based on a Petri net model," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 5, 658-669, 2001.
[35] N. Q. Wu and M. C. Zhou, *System Modeling and Control with Resource-oriented Petri Nets*, CRC Press, Taylor & Francis Group, New York, October 2009.
[36] N. Q. Wu and M. C. Zhou, "Process vs resource-oriented Petri net modeling of automated manufacturing systems," *Asian Journal of Control*, vol. 12, no. 3, 267-280, 2010.
[37] N. Q. Wu and M. C. Zhou, "A closed-form solution for schedulability and optimal scheduling of dual-arm cluster tools based on steady schedule analysis," *IEEE Transactions on Automation Science and Engineering,* vol. 7, no. 2, 303-315, 2010.
[38] 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 Transactions on Automation Science and Engineering*, vol. 9, no. 1, 203-209, Jan. 2012.
[39] 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 Transactions on Automation Science and Engineering*, vol. 9, no. 2, 446-454, Apr. 2012.
[40] N. Q. Wu, M. C. Zhou, L. P. Bai, and Z. W. Li, "Short-term scheduling of crude oil operations in refinery with high fusion point oil and two transportation pipelines," *Enterprise Information Systems*, vol. 10, no. 6, 581-610, May 2016.
[41] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Short-term scheduling of crude-oil operations: Petri net-based control-theoretic approach," *IEEE Robotics and Automation Magazine*, vol. 22, no. 2, 64-76, Jun. 2015.
[42] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Optimal one-wafer cyclic scheduling of time-constrained hybrid multicluster tools via Petri nets," *IEEE Transactions on Systems*, *Man*, *& Cybernetics*: *Systems*, vol. 47, no. 11, 2920-2932, Nov. 2017.
[43] F. J. Yang, N. Q. Wu, Y. Qao, M. C. Zhou, and Z. W. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Transactions on Systems*, *Man*, *& Cybernetics*: *Systems*, vol. 47, no. 3, 502-516, Mar. 2017.
[44] F. J. Yang, N. Q. Wu, Y. Q., M. C. Zhou, and R. Su, "Modeling and optimal scheduling of time-constrained single-robot-arm cluster tools via Petri nets and linear programming," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, DOI: 10.1109/TSMC.2017.2755599. 2017.
[45] F. J. Yang, N. Q. Wu, Y. Qiao, R. Su, "Polynomial approach to optimal one-wafer cyclic scheduling of treelike hybrid multi-cluster tools via Petri nets," *IEEE/CAA J. of Autom. Sinica*, vol. 5, no. 1, pp. 270-280, Jan. 2018.
[46] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Optimal one-wafer cyclic scheduling of hybrid multi-robot cluster tools with tree topology," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, vol. 48, no. 2, 289-298, 2018.
[47] F. J. Yang, N. Q. Wu, K. Z. Gao, C. J. Zhang, Y. T. Zhu, R. Su, and Y. Qiao, "Efficient approach to cyclic scheduling of single-arm cluster tools with chamber cleaning operations and wafer residency time constraint," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, 196-205, May 2018.
[48] J. Zhang, M. Khalgui, Z. Li, G. Frey, O. Mosbahi, and H. B. Salah, "Reconfigurable coordination of distributed discrete event control systems," *IEEE Trans. Control Syst. Technol*., vol. 23, no. 1, pp. 323-330, Jan. 2015.
[49] S. W. Zhang, N. Q. Wu, Z. W. Li, T. Qu, and C. D. Li, "Petri net-based approach to short-term scheduling of crude oil operations with less tank requirement," *Information Sciences*, vol. 417, 247-261, Nov. 2017.
[50] M. C. Zhou, C.-H. Wang, and X. Zhao, "Automating Mason's rule and its application to analysis of stochastic Petri nets," *IEEE Trans. Control Syst. Technol*., vol. 3, no. 2, pp. 238–244, Jun. 1995.
[51] M. C. Zhou and K. Venkatesh, *Modeling*, *Simulation and Control of Flexible Manufacturing Systems: A Petri net approach*, World Scientific, Singapore, 1998.
[52] Q. Zhu, N. Wu, Y. Qiao, and M. Zhou, "Scheduling of single-arm multi-cluster tools with wafer residency time constraints in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf*., vol. 28, no. 1, pp. 117-125, Feb. 2015.
[53] Q. H. Zhu, M. C. Zhou, Y. Qiao, and N. Q. Wu, "Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools," *IEEE Trans. Syst., Man, Cybern., Syst*., vol. 48, no. 3, 389-400, Mar. 2018.

Fajun Yang received the B. S. degree in Industrial Engineering from Hunan University of Science and Technology, Xiangtan, China, in 2011, the Ph. D. degree in Mechanical Engineering from Guangdong University of Technology, Guangzhou, China, in 2016. From 2015 to 2016, he was a Visiting Student with New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Research Fellow with Nanyang Technological University, Singapore. He has 10+ international journal papers (majority in the IEEE Transactions). His research interests are Petri nets, production planning, discrete event systems, and scheduling and control. He has served as a reviewer for a number of journals.

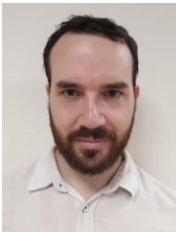
Yan Qiao (M'16) received the B. S. and Ph. D. degrees in Industrial Engineering and Mechanical Engineering from the Guangdong University of Technology, Guangzhou, China, in 2009 and 2015, respectively.

From 2014 to 2015, he was a Visiting Student with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA, and from 2016 to 2017, he is a Post-Doctoral Research Associate with the Institute of Systems Engineering, Macau University of Science and Technology (MUST), Taipa, Macau. He joined MUST in 2018. He is currently an assistant professor with the Institute of Systems Engineering, MUST, Taipa, Macau. He has one book chapter and 20+ international journal papers (majority in the IEEE Transactions). His research interests include discrete event systems, production planning, Petri nets, scheduling and control. He has served as a reviewer for a number of journals.



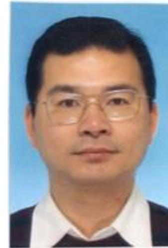
Kaizhou Gao received the B.Sc. and master degrees from China in 2005 and 2008 respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, 2016. From 2008 to 2012, he was with the School of Computer, Liaocheng University, Liaocheng, China. He was a research associate at the School of Electronic and Electrical engineering (EEE), NTU, Singapore, from Feb. 2012 to Sep.2013. From Oct. 2013 to Mar. 2015, he was a software engineer in Singapore Institute of Manufacturing Technology (SIMTech), A*star, Singapore. Since April 2015, he was a research fellow in the School of Electronic and Electrical engineering, NTU, Singapore. His research interests include intelligent computation, optimization, scheduling, and intelligent transportation. He has published over 50 refereed papers.

NaiQi Wu (M'04-SM'05) received his B. S. Degree in Electrical Engineering from Anhui University of Technology, Huainan, China, in 1982, the M. S. and Ph. D. Degrees in Systems Engineering both from Xi'an Jiaotong University, Xi'an, China in 1985 and 1988, respectively. From1988 to 1995, he was with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, and from 1995 to 1998, with Shantou University, Shantou, China. He moved to Guangdong University of Technology, Guangzhou, China in 1998. He joined Macau University of Science and Technology, Macau in 2013. He is currently a Professor at the Institute of Systems Engineering, Macau University of Science and Technology, Macau. His research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri net theory and applications, intelligent transportation systems, and energy systems. He is the author or coauthor of one book, five book chapters, and 130+ peer-reviewed journal papers. Dr. Wu was an associate editor of the IEEE Transactions on Systems, Man, & Cybernetics, Part C, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Systems, Man, & Cybernetics: Systems, and editor in chief of Industrial Engineering Journal. He is an associate editor of Information Sciences and IEEE/CAA Journal of Antomatica Sinica.



Ian Ware Simon received his B. S. degree in software development from Waikato University, New Zealand, in 2007, his Ph. D. degree in computer science from Waikato University in 2014. He is currently a Research Fellow with Nanyang Technological University, Singapore. His research interests are Discrete Event Systems, optimization, scheduling, and control. He has served as a reviewer for a number of journals.

Yuting Zhu received the B.S. degree from Southeast University, Jiangsu, China, in 2016. Currently, she is a Ph. D candidate in Nanyang Technology University, Singapore. Her research interests include discrete event systems and supervisory control.



Rong Su (M'11-SM'14) received the B.E. degree in automatic control from the University of Science and Technology of China, Hefei, China, in 1997, and the M.A.Sc. and Ph.D. degrees both in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2000 and 2004, respectively. Since then he was affiliated with University of Waterloo and Technical University of Eindhoven before he joined Nanyang Technological University, Singapore, in 2010. His research interests include discrete event systems, supervisory control, model-based fault diagnosis, multiagent systems, optimization and scheduling with applications in green buildings, flexible manufacturing, power management, and intelligent transportation systems. In the aforementioned areas, he has more than 130 publications in journals, book chapters, and conference proceedings, and two patents. Dr. Su is an Associate Editor for Journal of Discrete Event Dynamic Systems: Theory and Applications, Journal of Control and Decision, and Transactions of the Institute of Measurement and Control, and the Chair of IEEE Control Systems Society Technical Committee on Smart Cities.