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

Cyclic scheduling analysis of single-arm cluster tools with wafer residency time constraint and chamber cleaning operations

Yang, FaJun; Wu, NaiQi; Su, Rong; Qiao, Yan

2018

Yang, F., Wu, N., Su, R., & Qiao, Y. (2018). Cyclic scheduling analysis of single-arm cluster tools with wafer residency time constraint and chamber cleaning operations. Proceedings of 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), 241-246. doi:10.1109/coase.2018.8560461

https://hdl.handle.net/10356/140220

© 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/COASE.2018.8560461

Downloaded on 02 Apr 2023 08:15:14 SGT

Cyclic Scheduling Analysis of Single-arm Cluster Tools with Wafer Residency Time Constraint and Chamber Cleaning Operations

FaJun Yang, NaiQi Wu, *Senior Member*, *IEEE*, Rong Su, *Senior Member*, *IEEE*, Yan Qiao, *Member*, *IEEE*

***Abstract*-In semiconductor manufacturing, a cleaning operation that takes significant time is required for eliminating the chemical residual in a chamber after a wafer being processed and removed from it. Such a cleaning operation makes a traditional backward strategy inefficient. In the existing work, it is shown that the productivity can be improved if some of chambers at a step are kept empty. With this idea, an extended backward strategy is proposed by deciding the optimal number of empty chambers. Based on such a strategy, this work studies the challenging problem for scheduling a single-arm cluster tool with both chamber cleaning operations and wafer residency time constraint for the first time. By building timed Petri net model for the system, two linear programs are proposed to determine the minimal cycle time and test the existence of a feasible schedule. At last, an industrial example is used to demonstrate the obtained results.**

I. INTRODUCTION

Cluster tools are widely adopted for wafer processing in semiconductor fabrication. Generally, a cluster tool composes of a robot, two loadlocks, and several Process Chambers (PCs). Depending on the number of the arms for the robot, a tool is named single- or dual-arm cluster tool whose architecture is illustrated in Fig. 1.

Wafers to be processed enter a tool through the loadlocks in a cassette-by-cassette way. Then, they are transported to the PCs by the robot one by one for processing in a predefined sequence. After visiting all the processing steps, the wafers return to the cassette [15].

Some wafer manufacturing processes require that a processed wafer should be unloaded from a chamber in a limited time due to that the residual chemical gases and high temperature could damage its surface [10]. Such a requirement results in wafer residency time constraint (WRT constraint in short) and complicates the scheduling problem of cluster tools. For both dual- and single-arm cluster tools with WRT constraint, efficient ways are proposed in [11] [15] [16] to find a feasible and optimal cyclic schedule.

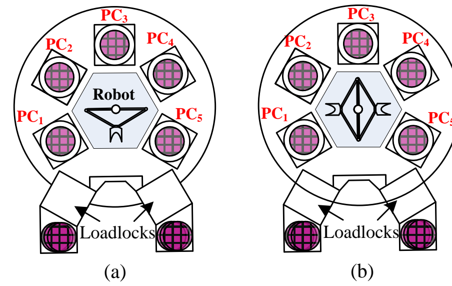

Fig. 1. (a) single-arm cluster tool and (b) dual-arm cluster tool.

Recently, to accommodate diversified customer needs, the wafer lot size is becoming smaller and smaller, leading to frequent lot switching operations [3] [7], and start-up and close-down processes of tools [4] [5] [9] [18].

As the wafer circuit line width continuously shrinks down, a stricter quality control is required for wafer fabrication processes. To do so, a chamber cleaning operation is needed after a chamber finishes processing m wafers. It is called "purge" if m = 1 [6] [8]. It is known that in a leading fab, purge is adopted for 50%~80% of the tools that are used for etching an chemical/physical vapor deposition.

In [17], by keeping some of chambers being idle, the authors propose an extended backward strategy and prove that, with this strategy, the globally optimal cycle time can be obtained in most practical cases and, even if it is not globally optimal, the gap between the lower bound of the global cycle time and the cycle time obtained by their proposed method is not too big. Furthermore, the extended backward strategy is quite simple and very close to the traditional backward strategy which is prevalently used in practice. Hence, in this work, the extended backward strategy is also adopted.

In practice, due to that the chamber cleaning operations could significantly increase the wafer sojourn time delays in a chamber such that it is challenging to make the WRT constraint satisfied. Motivated by this, based on the extended backward strategy proposed in [17], this work conducts a study on scheduling a single-arm cluster tool with both chamber cleaning operations and WRT constraint for the first time. Consequently, it presents an efficient method to test if a feasible schedule exists and find it if existing.

In comparison with the prior work, this work has the following primary contributions:

This work was supported in part by FDCT of Macau under Grant 106/2016/A3, in part by the National Natural Science Foundation of China under Grant U1401240, and in part by the Delta Electronics Inc and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

F. J. Yang and R. Su are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore, Singapore (e-mail: fjyang@ntu.edu.sg and RSu@ntu.edu.sg).

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).

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).

- Establish an efficient linear program to obtain the minimal system cycle time; and
- Derive an efficient linear program to test if there is a feasible schedule and find it if existing.

After a timed Petri net (PN) model is developed in Section II, Section III proposes two linear programs to calculate the system cycle time, check if a feasible schedule exists, and find it if existing. An example is given to demonstrate the power of the derived approach in Section IV. Finally, conclusions are summarized in Section V.

II. System Modeling

Assume that there are n processing steps for a single-arm cluster tool to process wafers. Let $\mathbf{\Omega}_n = \{0, 1, 2, \ldots, n\}$ and $\mathbb{N}_n = \mathbf{\Omega}_n \backslash \{0\}$. Then, for Step i, $i \in \mathbb{N}_n$, m_i is used to denote the number of parallel PCs at this step. By following Yu *et al.* [17], the number of empty chambers at Step i, $i \in \mathbb{N}_n$, is denoted by z_i. Thus, at Step i, at most m_i - z_i chambers are processing wafers at a time.

In [17], a method is proposed to determine z_i, $i \in \mathbb{N}_n$ and based on the determined z_i, an extended backward strategy is given. In this work, we assume that z_i, $i \in \mathbb{N}_n$, is determined and the resulting extended backward strategy is adopted.

To explain the meaning of the extended backward strategy, we first present the traditional backward strategy as the robot task sequence: ⟨unloads a finished wafer from the m_nth chamber at Step n → goes to the loadlocks and drops it there → goes to Step n-1 and unloads a completed wafer from its m_{n-1}th chamber → moves to Step n and drops it into the m_nth chamber → goes to Step n-2 and picks up a completed wafer from its m_{n-2}th chamber → … → goes to Step 1 and drops a wafer into the m_1th chamber → goes to Step n again and picks up a finished wafer from its (m_n-1)th chamber → goes to Step n-1 and unloads a completed wafer from its (m_{n-1}-1)th chamber …⟩. Note that, for a traditional backward strategy, after removing a processed wafer from the m_kth chamber at Step i, $i \in \mathbb{N}_n$, $1 \le m_k \le m_i$, the robot loads another wafer into the m_kth chamber within one cycle, and in the next cycle, it unloads a wafer from the (m_k-1)th (or m_ith if $m_k = 1$) chamber at Step i. However, for an extended backward strategy proposed in [17], after unloading a wafer from the m_kth chamber at Step i, the robot loads another wafer into the earliest emptied chamber at Step i rather than the m_kth chamber, unless $m_i = 1$. This is the difference between them.

With the above assumption, this work intends to answer whether there is a feasible schedule with WRT constraint and how to find it if existing.

A. Finite Capacity Petri Net

For modeling, analysis, and control of manufacturing systems, PNs are widely used as an efficient tool [11]-[13], [19-20]. Based on the work in [11], a PN is defined as follows.

Definition 1: A finite capacity PN is a directed graph with six sets of elements defined as follows.

1) P: a finite set of places.
2) T: a finite set of transitions, which is disjoint with P.
3) I: $(P \times T) \to \mathbb{N} = \{0, 1, 2, \ldots\}$ is an input function. If $I(p, t) > 0$, there is a directed arc from p to t with weight $I(p, t)$; if $I(p, t) = 0$, there is no such an arc.
4) O: $(P \times T) \to \mathbb{N}$ is an output function. If $O(p, t) > 0$, there is a directed arc from t to p with weight $O(p, t)$; if $O(p, t) = 0$, there is no such an arc.
5) $M(P) = (M(p_1), \ldots, M(p_m))^{\mathrm{T}}$ is a marking with M_0 being the initial one, where $M(p_i)$ denotes the number of tokens in p_i.
6) $\mathbb{K}$: a capacity function with $\mathbb{K}(p)$ representing the maximum number of tokens that p can hold at a time.

For a transition t, its preset is denoted by $^{\bullet}t = \{p: p \in P$ and $I(p, t) > 0\}$, ant its postset is represented by $t^{\bullet} = \{p: p \in P$ and $O(p, t) > 0\}$. Then, we can give the following definition [14].

Definition 2 [14]: In a finite capacity PN, a transition t is said to be enabled if $\forall p \in P$,

$$M(p) \ge I(p, t) \quad (2.1)$$

and

$$\mathbb{K}(p) \ge M(p) - I(p, t) + O(p, t) \quad (2.2)$$

When an enabled t fires at M, one can obtain a new marking

$$M'(p) = M(p) - I(p, t) + O(p, t) \quad (2.3)$$

By this definition, to enable t, we need enough tokens in $\forall p \in {^{\bullet}t}$ and enough free spaces in $\forall p \in t^{\bullet}$. Next, PN is used to model the system.

B. Petri Net for the System

With n processing steps (except the loadlocks) in the tool, we number them as Steps 1 to n. For concise presentation, the loadlocks are treated as Step 0 or Step n+1. By adopting an extended backward strategy, at Step i, $i \in \mathbb{N}_n$, there are z_i empty parallel chambers. Thus, at a time, at most z_i+1 parallel chambers are at the state of cleaning (it occurs for a time duration from the time instant when a wafer is removed from Step i to the time instant when another one is loaded into Step i) and, at most $e_i = m_i - z_i$ wafers are being processed. With such a fact, the PN of the system is modeled as follows.

Place p_i with $\mathbb{K}(p_i) = e_i = m_i$-$z_i$ models the PCs that are processing wafers at Step i, $i \in \mathbb{N}_n$, and p_0 the loadlocks with $\mathbb{K}(p_0) = \infty$. Places q_{i1} and q_{i2} with $\mathbb{K}(q_{i1}) = \mathbb{K}(q_{i2}) = 1$ model that the robot waits there before loading/unloading a wafer into/from p_i, $i \in \mathbf{\Omega}_n$. Place c_i with $\mathbb{K}(c_i) = z_i$+1 models the chambers that are being cleaned at Step i, $i \in \mathbb{N}_n$. As a chamber cleaning operation is not needed in the loadlocks, c_0 does not exist. Pictorially, a circle is used to denote these places. A special place r denoted by an ellipse is added to model the robot and we have $\mathbb{K}(r) = 1$, implying that the robot is single-armed.

Transition l_i models the loading activity of the robot at p_i, $i \in \mathbf{\Omega}_n$. The robot task of unloading a wafer from a chamber at p_i, $i \in \mathbf{\Omega}_n$, and moving to Step i+1 is modeled by transition u_i. With no wafer being carried, the robot moving from Steps i+2 to i is modeled by transition y_i, $i \in \mathbb{N}_{n-2}$, from Steps 0 to n-1 by y_{n-1}, and from Steps 1 to n by y_n. Pictorially, a bar is used to denote these transitions.

As no wafer can be dropped into a chamber before the chamber finishes its cleaning operation and after a wafer is

removed from a chamber, its cleaning operation starts immediately. Thus, arcs (c_i, l_i), (l_i, p_i), (p_i, u_i), and (u_i, c_i), $i \in \mathbf{\Omega}_n$, are added. Note that if $i = 0$, (c_0, l_0) and (u_0, c_0) does not exist. Then, by adding arcs (q_{i1}, l_i), (l_i, r), (r, y_i), (y_i, q_{i2}), (q_{i2}, u_i), and $(u_i, q_{(i+1)1})$, $i \in \mathbf{\Omega}_n$, we can obtain the PN structure for the system as shown in Fig. 2.

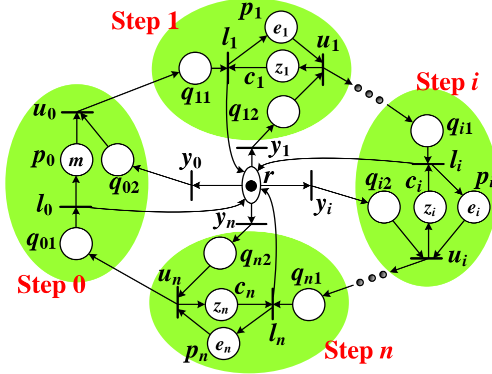

Fig. 2. PN model for system with n processing steps.

For the above obtained PN structure, we set its initial marking M_0 as follows: $M_0(q_{i1}) = M_0(q_{i2}) = 0$, $i \in \mathbf{\Omega}_n$; $M_0(p_i) = e_i = m_i - z_i$, $i \in \mathbb{N}_n$, and $M_0(p_0) = m$ to model the fact that, in the loadlocks, there are always wafers to be processed; $M_0(c_i) = z_i$, $i \in \mathbb{N}_n$; and $M_0(r) = 1$, as indicated by the number in the places shown in Fig. 2. By doing so, the PN model of the system is finally obtained.

With the PN model being built, we need to make sure that when the extended backward strategy is realized, the PN should be free of deadlock. To do so, the following control policy is presented.

Definition 3: For the above developed PN, at marking M, if $M(p_i) = m_i - z_i$, $i \in \mathbb{N}_n$, then transition y_n is control-enabled; and if $M(p_{i+1}) = m_i - z_i - 1$, $i \in \mathbb{N}_{n-1}$, then y_i is control-enabled.

By Definition 3, at M_0, only y_n is enabled, meanwhile, Conditions (2.1) and (2.2) are satisfied for y_n. Thus, it can fire and followed by u_n and then l_0, leading to marking M_1 such that only y_{n-1} is enabled due to that, after the firing of u_n, we have $M(p_n) = m_n - z_n - 1$. As Conditions (2.1) and (2.2) are also satisfied for y_{n-1}, it can fire and followed by u_{n-1} and then l_n. Next, y_{n-2} can fire. Consequently, after some time, y_0 can fire and followed by u_0 and then l_1. Then, marking M_n that is equivalent to M_0 is obtained, implying that a cycle is completed without deadlock, or the obtained PN model is deadlock-free.

C. Modeling Activity Time

To describe the temporal behavior of the system, both transitions and places are associated with time as follows since both of them model activities that take time. If a transition t is associated with time θ, firing t takes θ time units, while if a place p is associated with time θ, it implies that before a token in p can enable an output transition of p, it should stay in p for at least θ time units. By following [1], [2], we have:

1) The time for loading/unloading a wafer at a step is a known constant and denoted by λ; and
2) The time for robot moving between two steps with or without holding a wafer is a known constant and denoted by μ.

For Step i, $i \in \mathbb{N}_n$, let α_i and o_i denote the wafer processing and chamber cleaning time, δ_i the longest time for which a processed wafer can stay in a PC at Step i, and τ_i the wafer residency time (WRT) at Step i.

Further, let ω_{i1} and ω_{i2}, $i \in \mathbf{\Omega}_n$, represent the robot waiting time in q_{i1} and q_{i2}, $i \in \mathbf{\Omega}_n$. In Table I, we summarize the time taken for different transitions and places.

Table I. Time duration associated with transitions and places.

Symbol	Transition or place	Action	Time duration
λ	$l_i \in T$	Load a wafer into Step i, $i \in \mathbf{\Omega}_n$	λ
	$u_i \in T$	Unload a wafer from Step i and move to Step i+1, $i \in \mathbf{\Omega}_n$	$\lambda+\mu$
μ	$y_i \in T$	Move from Steps $i + 2$ to i, $i \in \mathbb{N}_{n-2}$	μ
	$y_{n-1} \in T$	Move from Steps 0 to n - 1	
	$y_n \in T$	Move from Steps 1 to n	
τ_i	$p_i \in P$	A wafer stays in a chamber of p_i, $i \in \mathbb{N}_n$	$[\alpha_i, \alpha_i+\delta_i]$
o_i	$p_i \in P$	A chamber at step i, $i \in \mathbb{N}_n$, is being cleaned	o_i
ω_{i1}	$q_{i1} \in P$	Wait before loading a wafer into Step i, $i \in \mathbf{\Omega}_n$	ω_{i1}
ω_{i2}	$q_{i2} \in P$	Wait before unloading a wafer from Step i, $i \in \mathbf{\Omega}_n$	ω_{i2}

With WRT constraint, a schedule obtained based on the evolution of the PN model may not be feasible. Hence, to obtain a feasible schedule, the following definition is presented.

Definition 4 [11]: Given the WRT interval $[\alpha_i, \alpha_i+\delta_i]$ for Step i, $i \in \mathbb{N}_n$, with the PN model, a schedule is feasible if whenever u_i starts to fire, $\alpha_i \leq \tau_i \leq \alpha_i + \delta_i$ holds.

III. System Scheduling

A. Temporal Properties

In this section, with the obtained timed PN, we analyze the robot cycle time, wafer cycle time, WRT, and so on such that a feasible schedule can be found.

It follows from the above analysis that, for both the traditional backward strategy and the extended one, the robot cycle time is identical if no robot waiting time is taken into account. Thus, by Wu *et al.* [11], for the extended backward strategy, without robot waiting, the robot cycle time is

$$\psi = 2(n + 1)(\mu + \lambda) \tag{3.1}$$

At Step i, $i \in \mathbb{N}_n$, to complete a wafer processing cycle, the robot activity sequence ⟨unloads a processed wafer from Step i and moves to Step i+1 with time $\lambda+\mu \to$ waits at $q_{(i+1)1}$ with time $\omega_{(i+1)1} \to$ loads that wafer into Step i+1 with time $\lambda \to$ moves to Step i-1 with time $\mu \to$ waits at $q_{(i-1)2}$ with time $\omega_{(i-1)2} \to$ unloads a processed wafer from Step i-1 and moves to Step i with time $\lambda+\mu \to$ waits at q_{i1} with time $\omega_{i1} \to$ loads it into Step i with time $\lambda \to$ a wafer is being processed at Step i with time $\alpha_i \to$ starts to unload from Step i again⟩ should be executed, which takes $\alpha_i + 4\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}$ time

units. Since m_i - z_i wafers are being processed concurrently at this step, the wafer processing cycle time at Step i, $i \in \mathbb{N}_n$, is

$$\zeta_i = (\alpha_i+4\lambda+3\mu+\omega_{(i+1)1}+\omega_{(i-1)2}+\omega_{i1})/(m_i-z_i),\ i \in \mathbb{N}_n \quad (3.2)$$

By removing $(\omega_{(i+1)1}+\omega_{(i-1)2}+\omega_{i1})$ from (3.2), we can obtain the allowed shortest wafer cycle time at Step i, $i \in \mathbb{N}_n$, as:

$$\xi_i = (\alpha_i + 4\lambda + 3\mu)/(m_i - z_i),\ i \in \mathbb{N}_n \quad (3.3)$$

Let Π denote the system cycle time. We present how to calculate the WRT τ_i at Step i, $i \in \mathbb{N}_n$, as follows. With the PN model shown in Fig. 2, we can deduce that, by the kth firing of l_i, the kth wafer is loaded into p_i. Then, by the $(k+ m_i-z_i)$th firing of u_i, this wafer is unloaded from p_i. During this time interval, (m_i-z_i) cycles are executed except that transitions u_i, l_{i+1}, y_{i-1}, u_{i-1}, l_i, and the robot waiting at $q_{(i+1)1}$, $q_{(i-1)2}$, and q_{i1} are executed $(m_i - z_i - 1)$ times. Hence, we have

$$\tau_i = (m_i - z_i)\times\Pi - (4\lambda+3\mu+\omega_{(i+1)1}+\omega_{(i-1)2}+\omega_{i1}),\ i \in \mathbb{N}_n \quad (3.4)$$

After unloading a wafer from a chamber (let c_k denote this chamber) at Step i, the cleaning operation of c_k starts immediately. Assume that κ_i time units later, the robot starts to drop another wafer into c_k. Then, we present how to calculate κ_i.

If $z_i = 0$, by the PN in Fig. 2, after unloading a wafer from c_k, the robot sequentially performs the following activities: ⟨moves to Step $i + 1$ with time $\mu \rightarrow$ waits in $q_{(i+1)1}$ with time $\omega_{(i+1)1} \rightarrow$ fires l_{i+1} with time $\lambda \rightarrow$ fires y_{i-1} with time $\mu \rightarrow$ waits in $q_{(i-1)2}$ with time $\omega_{(i-1)2} \rightarrow$ fires u_{i-1} with time $\lambda + \mu \rightarrow$ waits in q_{i1} with time ω_{i1} for loading another wafer into c_k⟩ which takes $2\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}$ time units. It means that $\kappa_i = 2\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}$.

If $z_i = 1$, after waiting in q_{i1}, the robot loads the wafer into the earliest emptied chamber at Step i and, in the next cycle, it loads a new wafer into c_k, which takes $\Pi + 2\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}$ time units, i.e., we have $\kappa_i = \Pi + (2\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1})$.

Similarly, if $z_i = 2$, we have $\kappa_i = 2\Pi + 2\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}$. Thus, for the general case, we have

$$\kappa_i = z_i\times\Pi + (2\lambda+3\mu+\omega_{(i+1)1}+\omega_{(i-1)2}+\omega_{i1}),\ i \in \mathbb{N}_n \quad (3.5)$$

With (3.1)-(3.5), we can analyze if a feasible schedule exists and present how to find it if existing.

B. *Schedulability Condition*

With WRT constraint, a critical issue is the existence of a feasible schedule. Before presenting the schedulability condition, we discuss how to calculate the system cycle time Π first.

By Wu *et al.* [11], we know that the robot cycle time with robot waiting time being included should be equal to Π. Thus, we have Constraint *a*): $\sum_{i=0}^{i=n}(\omega_{i1}+\omega_{i2}) + \psi = \Pi$, or $\sum_{i=0}^{i=n}(\omega_{i1}+\omega_{i2}) = \Pi - \psi$. Hence, to find a schedule for the system, the key is to allocate $\Pi - \psi$ to ω_{ij}'s, $i \in \mathbf{\Omega}_n$ and $j \in \mathbb{N}_2$. Once ω_{i1}'s and ω_{i2}'s, $i \in \mathbf{\Omega}_n$, are determined, a schedule is obtained. With Π being the system cycle time, ζ_i, $i \in \mathbb{N}_n$, should be less than or equal to Π. Therefore, by (3.2) and (3.3) we have that, for Step i, $i \in \mathbb{N}_n$, at most $(m_i - z_i) \times (\Pi - \xi_i)$ time units can be allocated to $(\omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1})$, or we have Constraint *b*): $\omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1} \leq (m_i - z_i) \times (\Pi - \xi_i)$, $i \in \mathbb{N}_n$.

For a chamber at Step i, $i \in \mathbb{N}_n$, no wafer can be loaded into it before its cleaning operation is finished. Thus, we have Constraint *c*): $\kappa_i \geq o_i$, where o_i denotes the time for cleaning the chamber and κ_i is determined by (3.5). Meanwhile, the WRT τ_i in a chamber at Step i, $i \in \mathbb{N}_n$, should be longer than or equal to the wafer processing time α_i, i.e., we have Constraint *d*): $\tau_i \geq \alpha_i$, where τ_i is determined by (3.4). Finally, we have Constraint *e*): $\omega_{i1} \geq 0$ and $\omega_{i2} \geq 0$, $i \in \mathbf{\Omega}_n$.

Note that, for Constraint *b*, it can be rewritten as $\omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1} + (m_i - z_i) \times \xi_i = \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1} + 4\lambda + 3\mu + \alpha_i \leq (m_i - z_i) \times \Pi$, or $\alpha_i \leq (m_i - z_i) \times \Pi - (\omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1} + 4\lambda + 3\mu) = \tau_i$, implying that Constraints *b* and *d* are identical. Thus, with minimizing Π as objective, Linear Program (3.6) is developed to obtain the minimal Π and set ω_{i1}'s and ω_{i2}'s, $i \in \mathbf{\Omega}_n$, as well.

$$\textit{Minimize } \Pi$$
$$\text{s.t.} \begin{cases} \sum_{i=0}^{i=n}(\omega_{i1}+\omega_{i2}) = \Pi - \psi \\ \omega_{(i+1)1}+\omega_{(i-1)2}+\omega_{i1} \leq (m_i - z_i)\times(\Pi - \xi_i),\ i \in \mathrm{N}_n \\ o_i \leq z_i \times \Pi + (2\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}),\ i \in \mathrm{N}_n \\ \omega_{i1} \geq 0,\ \omega_{i2} \geq 0,\ i \in \Omega_n \end{cases} \quad (3.6)$$

With the minimal Π being determined by solving (3.6), we need to check if the WRT constraint is met. By Definition 4, for a feasible schedule, $\tau_i \leq \alpha_i + \delta_i$ has to be satisfied, i.e., $(m_i - z_i) \times \Pi - (4\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}) \leq \alpha_i + \delta_i$, $i \in \mathbb{N}_n$. Then, based on (3.6), we present Linear Program (3.7) with minimizing Θ as objective to check if a feasible schedule exists, where $\Theta \geq \Pi$.

$$\textit{Minimize } \Theta$$
$$\text{s.t.} \begin{cases} \Theta \geq \Pi \\ \sum_{i=0}^{i=n}(\omega_{i1}+\omega_{i2}) = \Theta - \psi \\ \omega_{(i+1)1}+\omega_{(i-1)2}+\omega_{i1} \leq (m_i - z_i)\times(\Theta - \xi_i),\ i \in \mathrm{N}_n \\ o_i \leq z_i \times \Theta + (2\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}),\ i \in \mathrm{N}_n \\ (m_i - z_i)\times\Theta\text{-}(4\lambda + 3\mu + \omega_{(i+1)1} + \omega_{(i-1)2} + \omega_{i1}) \\ \qquad \leq \alpha_i + \delta_i,\ i \in \mathrm{N}_n \\ \omega_{i1} \geq 0,\ \omega_{i2} \geq 0,\ i \in \Omega_n \end{cases} \quad (3.7)$$

By (3.7), if a solution is found by determining ω_{i1}'s and ω_{i2}'s, $i \in \mathbf{\Omega}_n$, it must be feasible. On the contrary, if there is no solution for (3.7), we cannot find a feasible schedule, or the WRT constraint is violated with the extended backward strategy.

By using a commercial solver, a linear program can be efficiently solved. Thus, the proposed approach is computationally efficient. Since the WRT constraint is embedded into (3.7), we have the following schedulability result immediately.

Theorem 3.1: For a single-arm cluster tool with chamber cleaning operation and WRT constraint, by using the extended backward strategy, if a solution can be found by

solving Linear Program (3.7), the obtained schedule is feasible.

It should also be pointed out that the cycle time Θ for the obtained schedule is minimized by Linear Program (3.7), the maximal productivity is achieved for a determined extended scheduling strategy.

C. Experimental Performance of Proposed method

Next, we randomly generate some problem instances to compare the gap between the lower bound of the global cycle time and the cycle time obtained by the proposed method in this work.

Let Θ^* denote the lower bound of the global cycle time. In step i, $i\in\mathbb{N}_n$, after a wafer finishes its processing (taking α_i time units), the robot removes this wafer from Step i (taking λ time units), after that, the cleaning operation begins (o_i) immediately. From the starting time of cleaning operation at Step i, it takes at least $2\lambda + 3\mu$ time units for the robot to load another wafer into Step i, that is: moving to the next step (μ) $\rightarrow$ loading a wafer there (λ) $\rightarrow$ moving to step i - 1 (μ) $\rightarrow$ unloading another wafer $\rightarrow$ moving to step i again ($\lambda + \mu$). If the cleaning operation finished, then, loading that wafer into Step i (λ).

From the above analysis we can get that the workload at Step i is larger than or equal to $[\alpha_i + 2\lambda + \max\{o_i, 2\lambda + 3\mu\}]/m_i$, where m_i denotes the number of parallel chambers at Step i. Hence, Θ^* can be calculated by the following equation,

$$\Theta^* = Max\left\{ {\left[\alpha_i + 2\lambda + max\{o_i,\ 2\lambda + 3\mu\}\right]}\big/{m_i} \right\},\ i\in\mathbb{N}_n \quad (3.8)$$

With Θ^*, we define the relative optimality gap between Θ^* and Θ as

$$\text{Gap (\%)} = \frac{\Theta\text{-}\Theta^*}{\Theta^*}\times 100, \quad (3.9)$$

Next, we use a case to show the variation of the gap.

A cluster tool has three processing steps and its configuration is $(m_1, m_2, m_3) = (2, 2, 1)$ and $(z_1, z_2, z_3) = (1, 0, 0)$. The robot loading/unloading takes six time units and, the robot moving with or without holding a wafer takes two time units, or we have $\lambda = 6$ and $\mu = 2$.

For Step 1, wafer processing time $\alpha_1 = 120$ and chamber cleaning time $o_1 = 140$;

For Step 3, $\alpha_3 = 60$ and $o_3 = 18$;

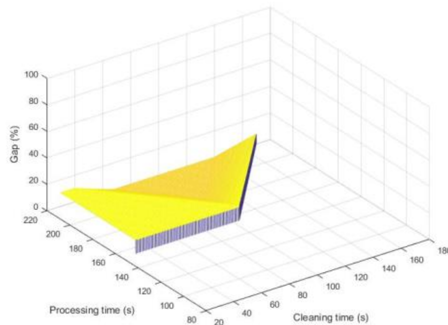

Fig. 3. Experimental optimality gap for case 1.

For Step 2, we have $\alpha_2 \in [80, 220]$, and $o_2 \in [30, 180]$, note that these ranges are most common in etching and chemical vapor deposition steps. After being processed, a wafer can stay at each step for no more than 30 time units, or $\delta_1 = \delta_2 = \delta_3 = 30$. The experimental optimality gap is shown in Fig. 3.

In Fig. 3, the area with no color means that there is no feasible solution or the WRT constraint is violated for the corresponding processing and cleaning times. This is because that Step 2 has two parallel chambers to process wafer at the same time ($m_2 = 2$ and $z_2 = 0$). Hence, if the wafer processing time is too small, it could violate the WRT constraint easily. Also, if both the cleaning and wafer processing times are very large, the system could also violate the WRT constraint easily due to that the system cycle time could be too large.

For the area with yellow color, we know that about half of them has a zero gap, and for the remaining part, the gap is very small, about 8%.

Actually, the proposed method is globally optimal in most practical cases and the relative gap is also not very big even if it is not globally optimal. Furthermore, such a scheduling strategy is simple and very close to the traditional backward strategy, thus, it is widely applicable in the practice.

IV. Illustrative Example

A cluster tool has three processing steps. There are three Process Chambers (PCs) at Step 1, and two PCs at Steps 2 and 3, or we have $(m_1, m_2, m_3) = (3, 2, 2)$. An extended backward strategy is adopted by keeping one chamber empty at each step, or we have $(z_1, z_2, z_3) = (1, 1, 1)$.

$$\textit{Minimize } \Pi$$
$$\text{s.t.} \begin{cases} \omega_{01} + \omega_{02} + \omega_{11} + \omega_{12} + \omega_{21} + \omega_{22} + \omega_{31} + \omega_{32} = \Pi\text{-}56 \\ \omega_{21} + \omega_{02} + \omega_{11} \le 2\times(\Pi - 83) \\ \omega_{31} + \omega_{12} + \omega_{21} \le \Pi - 86 \\ \omega_{01} + \omega_{22} + \omega_{31} \le \Pi - 116 \\ 120 \le \Pi + (16 + \omega_{21} + \omega_{02} + \omega_{11}) \\ 100 \le \Pi + (16 + \omega_{31} + \omega_{12} + \omega_{21}) \\ 120 \le \Pi + (16 + \omega_{01} + \omega_{22} + \omega_{31}) \\ \omega_{01}, \omega_{02}, \omega_{11}, \omega_{12}, \omega_{21}, \omega_{22}, \omega_{31}, \omega_{32} \ge 0 \end{cases} \quad (4.1)$$

For a PC at Steps 1, 2, and 3, it takes 140, 60, and 90 time units to process a wafer, 120, 100, and 120 time units to be cleaned; five time units for the robot's unloading/loading at a step; two time units for the robot's moving between PCs with or without holding a wafer. After being processed, a wafer can stay in a PC for no more than 20 time units. In other words, we have $\alpha_1 = 140$, $\alpha_2 = 60$, $\alpha_3 = 90$; $o_1 = 120$, $o_2 = 100$, $o_3 = 120$, $\lambda = 5$, $\mu = 2$, and $\delta_1 = \delta_2 = \delta_3 = 20$.

For this case, we have:

$\psi = 2(n + 1)(\mu + \lambda) = 8 \times 7 = 56$
$\xi_1 = (\alpha_1 + 4\lambda + 3\mu)/(m_1 - z_1) = (140 + 26)/2 = 83$
$\xi_2 = (\alpha_2 + 4\lambda + 3\mu)/(m_2 - z_2) = 86$
$\xi_3 = (\alpha_3 + 4\lambda + 3\mu)/(m_3 - z_3) = 116$

Then, by (3.6), to calculate the optimal cycle time, we have Linear Program (4.1).

By solving (4.1), we have $\Pi = 116$. Then, based on the

decided Π and (3.7), we have Linear Program (4.2) to check if there is a feasible schedule. By solving (4.2), a feasible schedule is obtained with $\Theta = \Pi = 116$, $\omega_{02} = 50$, $\omega_{21} = 10$, and $\omega_{01} = \omega_{11} = \omega_{12} = \omega_{22} = \omega_{31} = \omega_{32} = 0$. The correctness of the obtained schedule can be verified by simulation.

$$\text{Minimize } \Theta$$
$$\text{s.t.} \begin{cases} \Theta \geq 116 \\ \omega_{01}+\omega_{02}+\omega_{11}+\omega_{12}+\omega_{21}+\omega_{22}+\omega_{31}+\omega_{32}=\Theta\text{-}56 \\ \omega_{21}+\omega_{02}+\omega_{11} \leq 2\times(\Theta-83) \\ \omega_{31}+\omega_{12}+\omega_{21} \leq \Theta-86 \\ \omega_{01}+\omega_{22}+\omega_{31} \leq \Theta-116 \\ 120 \leq \Theta+(16+\omega_{21}+\omega_{02}+\omega_{11}) \\ 100 \leq \Theta+(16+\omega_{31}+\omega_{12}+\omega_{21}) \\ 120 \leq \Theta+(16+\omega_{01}+\omega_{22}+\omega_{31}) \\ 2\times\Theta-26-(\omega_{21}+\omega_{02}+\omega_{11}) \leq 160 \\ \Theta-26-(\omega_{31}+\omega_{12}+\omega_{21}) \leq 80 \\ \Theta-26-(\omega_{01}+\omega_{22}+\omega_{31}) \leq 110 \\ \omega_{01},\omega_{02},\omega_{11},\omega_{12},\omega_{21},\omega_{22},\omega_{31},\omega_{32} \geq 0 \end{cases} \quad (4.2)$$

V. CONCLUSIONS

With the shrink down of wafer circuit widths, stringent quality control is desired for wafer processing such that, in some leading labs, a chamber cleaning operation is required after each wafer is removed from a chamber. By considering such operations, to obtain an efficient schedule, Yu *et al*. [17] propose an extended backward strategy for single-arm cluster tools by keeping some numbers of chambers empty at a processing step. However, no wafer residency time constraint is considered in their work. This work represents the first one which takes both chamber cleaning operations and wafer residency time constraint into consideration. First, it develops a timed Petri net model to describe the dynamic characteristic of the system. Based on it, two linear programs are formulated to calculate the minimum cycle time and test if there is a feasible schedule.

In this work, a chamber cleaning operation is required after a chamber finishes processing just $m = 1$ wafer. So, it is very meaningful to extend the result of this work to the cases with $m > 1$. Besides, the activity time is treated as known constant in this work. Actually, they could be subject to random variation, resulting in residency time fluctuation in a chamber, which makes the scheduling problem more challenging.

REFERENCES

[1] 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, pp. 84-89, 2004.

[2] 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, pp. 521-534, 2003.

[3] 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.

[4] H. J. Kim, J. H. Lee, and T. E. Lee, "Noncyclic Scheduling of Cluster Tools With a Branch and Bound Algorithm," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 690-700, 2015.

[5] H. J. Kim, J. H. Lee, and T. E. Lee, "Schedulability analysis for noncyclic operation of time-constrained cluster tools with time variation," *IEEE Transactions on Automation Science and Engineering*, vol.13, no. 3 , pp. 1409-1414, 2016.

[6] H. Kim, H. J. Kim, J. H. Lee, and T. E. Lee, "Scheduling dual-armed cluster tools with cleaning processes," *Int. J. Prod. Res*., vol. 51, no. 12, pp. 3671–3687, 2013.

[7] J. H. Lee and H. J. Kim, "Makespan analysis of lot switching period in cluster tools", *IEEE Transactions on Semiconductor Manufacturing*, vol. 29, no.2, pp. 127-136, 2016.

[8] S. M. Noh, J. H. Kim, and S. Y. Jang, "Scheduling of single-armed cluster tool considering cleaning processes," in *Proc. Asia Simulation Conf.*, pp. 475–484, 2015.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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, pp. 196-207, 2013.

[14] 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, pp. 658-669, 2001.

[15] 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, pp. 303-315, 2010.

[16] F. J. Yang, N. Q. Wu, Y. Q., M. C. Zhou, and R. Su. "Modeling and optimal scheduling of wafer-residency-time constrained cluster tools via Petri nets and linear programming," *IEEE Transactions on Systems, Man and Cybernetics: Systems*. DOI: 10.1109/TSMC.2017.2755599, 2017.

[17] T. S. Yu, H. J. Kim, and T. E. Lee, "Scheduling Single-Armed Cluster Tools With Chamber Cleaning Operations," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 705-716, Apr. 2018.

[18] 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*., online, vol. 48, no. 3, pp. 389-400, Mar. 2018.

[19] 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.

[20] 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.