Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-7 (2018) 251–256

Scheduling and Control of Start-up Process 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***

***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).

****Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau**

(e-mail: nqwu@must.edu.mo, yqiao@must.edu.mo)

***Abstract*-In wafer manufacturing, with the shrinking down of wafer lot size, cluster 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. In the existing work, optimal scheduling of start-up process for time-constrained single-arm cluster tools has been addressed under the assumption that each processing step consists of just one process chamber. This work relaxes this strict restriction by treating that multiple process chambers could be configured for processing steps. By building Petri net model for the start-up process, a linear program is derived to search a feasible schedule with minimal makespan for time-constrained single-arm cluster tools with parallel chambers for the first time. One industrial example is given to demonstrate the effectiveness of the obtained results.**

Keywords: **Cluster tools, wafer manufacturing, transient process, Petri net, scheduling.**

I. INTRODUCTION

As a kind of computer-controlled complex equipment, cluster tools, as shown in Fig 1, are formed by several process chambers (PCs), two loadlocks, and one robot.

Wafer residency time constraint (WRTC) is critical for many wafer fabrication processes, which requires that a processed wafer in a PC can reside there for no more than a given time and its violation could result in a quality problem [3, 14]. A cluster tool with such a constraint is called a time-constrained one.

Much effort has been done for scheduling cluster tools with WRTC [3, 9, 14, 15, 18-21]. In [15, 18], for time-constrained single and dual-arm cluster tools, a feasible schedule can be easily obtained by analytical expressions if the schedulability conditions are satisfied. In [20], two linear programs are developed to find a feasible schedule for single-arm cluster tools. In comparison with [15] where the scheduling strategy is limited to be backward, the strategy in [20] can be arbitrary.

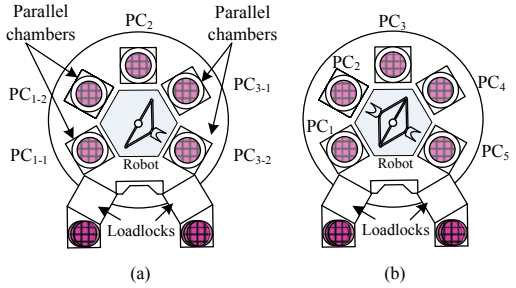


Fig. 1. (a) single-arm cluster tool; (b) dual-arm cluster tool.

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 [4-8, 13, 22]. Thus, scheduling a transient process becomes an important issue to be addressed.

In [13, 22], 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, respectively. However, only one PC is configured for each processing step in their work. In practice, to balance the workloads, multiple PCs need to be configured in parallel for a processing step.

Compared with [13, 22], this work addresses scheduling problem of start-up process of single-arm cluster tools where there are multiple PCs for some steps and the wafer residency time for different PCs for the same step can be different. Thus, the addressed situations are more practical and challenging.

In comparison with the existing work, this work makes the following primary contributions:

1) Establish for the first time a timed PN to model the start-up process of single-arm cluster tools with parallel chambers; and
2) Derive a linear program to obtain a feasible schedule with minimal makespan for the start-up process.

In the next Section, a timed PN model is developed for the start-up process. Based on it, Sections III derives a linear program to obtain an optimal and feasible schedule for the start-up process. An example is used to demonstrate the effectiveness of the proposed approach in Section IV. Finally, conclusions are summarized in Section V.

II. SYSTEM MODELING AND CONTROL

Let $\mathbf{N}_n = \{1, 2, \ldots, n\}$ and $\mathbf{\Omega}_n = \mathbf{N}_n \cup \{0\}$. For a single-arm cluster tool, we assume that it consists of n processing steps (except the loadlocks) and a step may be composed of m_i ($m_i \geq 1$) PCs. Thus, a wafer flow pattern can be indicated by $(m_1, m_2, \ldots, m_n)$. For concise presentation, the loadlocks

10.1016/j.ifacol.2018.06.309

are treated both as Step 0 and Step n + 1 with no processing function.

For a single-arm cluster tool in the steady state, a backward strategy is optimal [10] since in most cases, the wafer processing time is much longer than the robot activity time. For the start-up process, Kim *et al*. [6] prove that a generalized backward strategy can obtain a schedule to achieve a minimal makespan in most cases. Hence, such strategies are adopted for the steady state and start-up process in this work. As backward strategy is well known in this field, we just explain how it is generalized for the start-up process in the follows.

2.1 Generalized Backward Strategy for Start-up Process

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

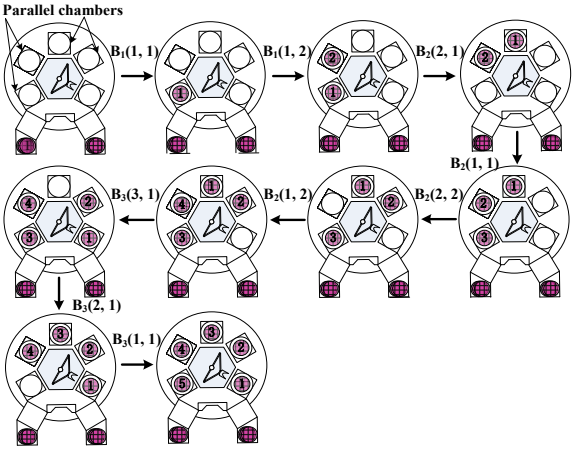


Fig. 2. Generalized backward strategy for start-up process.

Define a basic backward operation A_i for Step i with A_i = ⟨unloading a wafer from Step i - 1 → moving to Step i → loading it into Step i⟩, $i \in \mathbf{N}_{n+1}$. 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, $d \in \mathbf{N}_n$. 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 follows:

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

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

2.2 Finite Capacity Petri Net

As an efficient method for modeling, analysis, and control, PNs are widely applied in discrete manufacturing systems [2-5, 11-12, 15-16, 23-26]. Following the work in [15, 17], their basic definition 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 $P \cap T = \varnothing$ and, $P \cup T \ne \varnothing$.
3) I: $(P \times T) \to \mathbf{N} = \{0, 1, 2, \ldots\}$ is an input function specifying the existence of a directed arc from place p to transition t with weight $I(p, t)$.
4) O: $(P \times T) \to \mathbf{N}$ 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) K: a capacity function with $K(p)$ denoting the maximum number of tokens that p can accommodate.

Details about the transition enabling and firing rules of PNs can be found in [15, 17]. Next, we use the finite capacity PN to model the system.

2.3 Modeling the Steady State and Start-up Processes

With a backward strategy, the PN model for a single-arm tool under the steady state is developed in [20] and it is depicted in Fig. 3.

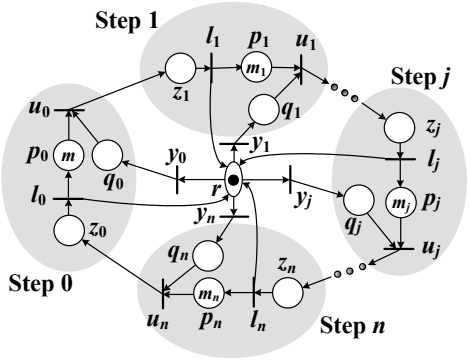


Fig. 3. PN model for the steady state of a single-arm cluster tool.

In Fig. 3, p_0 models the loadlocks with $K(p_0) = m$ representing that there are always wafers to be processed in the loadlocks. For Step j, $j \in \mathbf{N}_n$, place p_j with $K(p_j) = m_j$ models the step and indicates that there are m_j PCs. For each Step j, $j \in \mathbf{\Omega}_n$, place q_j with $K(q_j) = 1$ 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 $K(z_j) = 1$ represents that the robot rotates from Step j-1 to j with a wafer being held. Note that both Step 0 and Step 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 modelled by place r with $K(r) = 1$.

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 different step to j with no wafer being held is modelled by y_j, $j \in \mathbf{\Omega}_n$. Note that y_j, $j \in \mathbf{N}_n$, can also denote the robot moving at Step j from its one chamber to another with no wafer being held. Then, by adding arcs (u_{j-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 in the steady state as shown in Fig. 3.

For the start-up process, when the tool is started for

operation, the robot acts according to task sequence: ⟨unloading wafer 1 from one 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 one loadlock if $m_1 \geq 2$ → moving to Step 1 → loading it there → … → unloading wafer m_1 from one loadlock → moving to Step 1 → loading it there⟩ such that state M_1 is reached.

During the evolution process from $M_{d\text{-}1}$ to M_d, $d \in \mathbf{N}_n \backslash \{1\}$, the robot follows the task sequence: ⟨moving to Step d - 1 → unloading a wafer there → moving to Step d → loading that wafer there → moving to Step d - 2 → unloading a wafer there → moving to Step d - 1 → loading that wafer there → … → moving to the loadlocks → unloading a wafer there → moving to Step 1 → loading that wafer there⟩. By repeating this process m_d times, M_d is reached. Finally, when M_n is reached, 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 $M_{d\text{-}1}$ to M_d, $d \in \mathbf{N}_n$, is shown in Fig. 4, where the meaning of the places and transitions is identical to that in Fig. 3. Note that, in Fig. 4, the marking in Step d is k, $0 \leq k < m_d$.

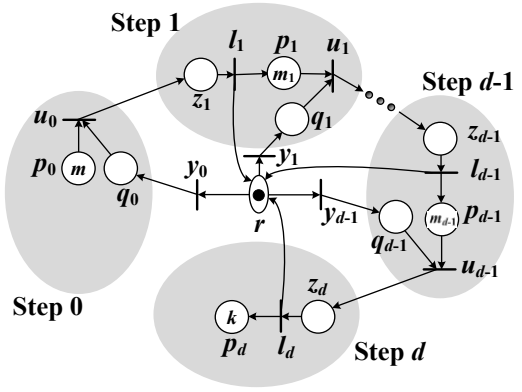


Fig. 4. PN model for the start-up process of a single-arm cluster tool from state $M_{d\text{-}1}$ to M_d, $d \in \mathbf{N}_n$.

With the PN models in Figs. 3-4, we need to ensure that the backward strategy and generalized backward strategy can be realized with no deadlock. To do so, for the steady state, by following [15], a control policy is given as follows.

Definition 2: For the PN shown in Fig. 3, at any marking M, transition y_n is said to be control-enabled if $M(p_i) = m_i$, $i \in \mathbf{N}_n$; and y_i, $0 \leq i < n$, is said to be control-enabled if $M(p_{i+1}) = m_{i+1} - 1$.

By Wu *et al*. [15], with Definition 2, the PN in Fig. 3 is deadlock-free in the steady state. Next, for the start-up process, a control policy is given as follows.

Definition 3: For the PN shown in Fig. 4, 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 $M(p_k) = m_k$, $k \in [1, i]$, and $M(p_{i+1}) < m_{i+1}$.

By Definition 3, when the tool begins to work, $M(p_1) = 0 < m_1$, y_0 is control-enabled and it repeats to fire until $M(p_1) = m_1$ such that state M_1 is reached. Then, y_1 is control-enabled due to that $M(p_2) = 0 < m_2$ 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. In this way, we can reach state M_2. By continuing this process, we finally reach M_n, which means that the generalized backward strategy for the start-up process can be realized with no deadlock.

2.4 Modeling Activity Time

To describe the temporal behavior of the PN models shown in Figs. 3-4, we associate both transitions and places with time. By following [1, 3], we 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 chambers with or without carrying a wafer is also a deterministic constant and denoted by μ.

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, $i \in \mathbf{N}_{n+1}$	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, $i \in \mathbf{N}_n$, with no wafer being held	μ
τ_i	$p_i \in P$	A wafer stays in a chamber of p_i, $i \in \mathbf{N}_n$	$[\alpha_i, \alpha_i+\delta_i]$

Let α_i represent the wafer processing time at Step i, $i \in \mathbf{N}_n$. After being processed, a wafer can stay in a chamber for at most δ_i, $i \in \mathbf{N}_n$, time units. Let τ_i represent the wafer residency time at Step i, $i \in \mathbf{N}_n$. The time associated with different transitions and places is summarized in Table I. Consider the WRTC, a feasible schedule is defined as follows.

Definition 5 [15]: In the PN models shown in Figs. 3-4, if Step i, $i \in \mathbf{N}_n$, has the WRTC interval $[\alpha_i, \alpha_i+\delta_i]$ and whenever u_i starts to fire, $\alpha_i \leq \tau_i \leq \alpha_i + \delta_i$ holds, then, a 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, $i \in \mathbf{N}_n$, 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 know ϕ_1 and ϕ_2. To do so, for the start-up process, we 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 $M_{d\text{-}1}$ to M_d, $d \in \mathbf{N}_n$, $i \in \mathbf{\Omega}_{n\text{-}1}$, and $1 \leq j \leq m_d$.
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, $d \in \mathbf{N}_n$, $i \in \mathbf{N}_n$, 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, $d \in \mathbf{N}_n$, $i \in \mathbf{\Omega}_{n-1}$, and $1 \le j \le m_d$.

Then, during the process from M_0 to M_1, we let

$$t_1(0, 1, 2) = \omega_1(0, 1) = 0 \quad (3.1)$$

By (3.1), it means that the initial time is set to zero, at which the robot starts to unload the first wafer from the loadlocks. It follows from Table I 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

$$t_1(1, 1, 1) = t_1(0, 1, 2) + 2\lambda + \mu \quad (3.2)$$

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 the loadlocks and unload another wafer. In this case, we have:

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

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

In this way, state M_1 is reached. Next, the model shown in Fig. 4 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. This task takes μ time units. Also, if $m_1 = 1$ and $m_2 \ge 2$, during the process from state M_1 to M_2, after performing $t_2(1, j, 1)$, $1 \le j \le m_2\text{-}1$, the robot needs to wait at Step 1 until this wafer 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 state M_0 to M_1, with the generalized backward strategy, the system evolves from state M_{d-1} to M_d, $d \in \mathbf{N}_n \backslash \{1\}$, 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 $m_1 = 1$,

$$t_2(1, 1, 2) = t_1(1, 1, 1) + \omega_2(1, 1) \quad (3.5)$$

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

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

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

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

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

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 is ended. 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 this wafer finishes its processing, and then, unloads this wafer, which means that no robot moving is required. Similarly, for (3.5), if $m_1 = 1$, no robot moving is required.

After $t_n(1, m_n, 1)$ is executed (loading a wafer into Step 1), there is one wafer in processing in each chamber such that state M_n (the steady state) is reached. Starting from the end time of performing $t_n(1, m_n, 1)$, we use c_1 to denote the first steady cycle that ends at the time when a wafer has just loaded into Step 1. 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, $d \ge 1$.
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, $d \ge 1$, and
3) $\omega_{n+d}(i, 1)$ denote the robot waiting time before unloading a wafer from Step i during steady cycle c_d, $d \ge 1$.

Note that Step 0 equals Step n+1. During the start-up process, to complete the process as soon as possible, the robot waiting time at Step i, $i \in \mathbf{\Omega}_n$, could be scheduled to be different. However, in 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 $\mathcal{R} = \max\{m_i | i \in \mathbf{N}_n\}+1$. Then, similar to (3.5)-(3.8) we have:

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

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

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

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

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

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

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

Since the robot waiting time should be nonnegative, we have

$$\omega_{n+j}(m, 1) \ge 0,\ j \in \mathbf{N}_{\mathcal{R}} \quad (3.15)$$

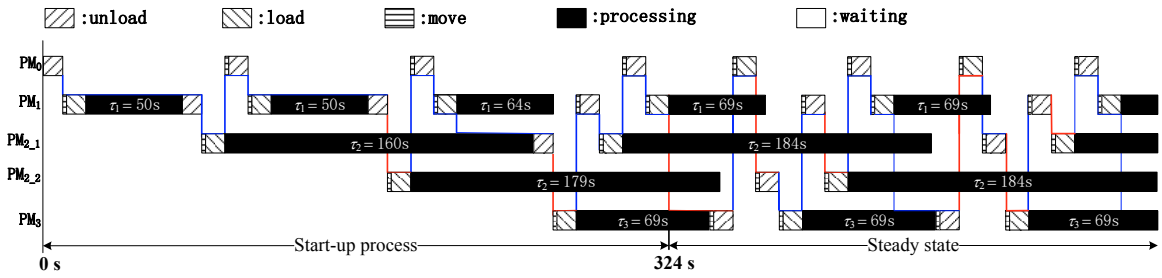
$$\omega_d(d\text{-}m, j) \ge 0,\ 1 \le d \le n,\ 1 \le m \le d,\ 1 \le j \le m_d \quad (3.16)$$

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.

In order to calculate the wafer sojourn time in a chamber, we assume that a wafer is loaded into Step i by executing $t_d(i, j, 1)$, where the values of d, i and j are deterministic. If $\sum_{k=i}^{n} M(p_k) \le \sum_{l=i+1}^{n} m_l$, where $M(p_k)$ represents the number of tokens (wafers) in p_k after executing $t_d(i, j, 1)$, we assume that this 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 $\sum_{l=i+1}^{q-1} m_l < \sum_{k=i}^{n} M(p_k) \le \sum_{l=i+1}^{q} m_l$;
2) $z = \sum_{k=i}^{n} M(p_k) - \sum_{l=i+1}^{q-1} m_l$.

With $\sum_{l=i+1}^{q-1} m_l < \sum_{k=i}^{n} M(p_k) \le \sum_{l=i+1}^{q} m_l$, we can conclude



that this wafer is unloaded from a PC at Step i during the state evolution process from M_{q-1} to M_q, and it is the $(\sum_{k=i}^{n} M(p_k) - \sum_{l=i+1}^{q-1} m_l)$th one unloaded from Step i during this state evolution process.

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

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

However, if $\sum_{k=i}^{n} M(p_k) > \sum_{l=i+1}^{n} m_l$, we can easily get that this wafer is unloaded by performing $t_{n+r}(i, 1, 2)$, where $r = \sum_{k=i}^{n} M(p_k) - \sum_{l=i+1}^{n} m_l$. Hence, we have the following constraint.

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

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

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

Subject to: (3.1)-(3.18).

In LP1, the objective is to minimize the total robot waiting time. In practice, a cluster tool usually consists of 6-8 PCs such that the above linear program can be 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.

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

With Theorem 3.1, the remaining question is whether an optimal and feasible solution can be found for any tool by LP1. To answer this question, we introduce the virtual wafer method which is proposed by Wu *et al*. [15] for the start-up process.

During the start-up process, with the virtual wafer method, it is assumed that there is one virtual wafer in each chamber at the beginning such that, during the process from the idle state to the steady state, the tool operates with the way that is adopted under the steady state (the schedule under the steady state can be obtained according to Wu *et al*. [15]). In this way, each time when the robot needs to unload a wafer from the loadlocks, it unloads a real one and, when it requires to load a wafer into the loadlocks, 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, during the state evolution process from M_d to M_{d+1}, $1 \le d \le n - 1$, by treating the time taken for performing virtual operations as waiting time at Step d, we have the following result directly.

Theorem *3.2*: An optimal and feasible schedule for the start-up process can be obtained by LP1 if there is a feasible periodic schedule for a single-arm cluster tool with WRTC under the steady state.

IV. ILLUSTRATIVE EXAMPLE

A cluster tool has a wafer flow pattern (1, 2, 1). For a PC at Steps 1, 2, and 3, it takes 50, 160, and 69 time units to process a wafer and, after being processed, a wafer can stay there for at most 20, 26 and 15 time units. It needs 10 time units to do unloading/loading at each step, 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 $\mu = 2$.

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

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

For this example, by Wu *et al*. [15], a feasible schedule with cycle time 115 is obtained and the robot waiting time is set as: $w_0 = w_1 = w_2 = 0$, and $w_3 = 19$, where w_i, $i \in \mathbf{\Omega_3}$, denotes the robot waiting time at Step i in the steady state. With the virtual wafer method proposed by Wu *et al*. [15], 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, $\omega_4(0, 1) = \omega_5(0, 1) = \omega_6(0, 1) = 0$.

Its simulation result is shown in Fig. 5 and it takes 324 time units to complete the start-up process. Compared with the result obtained by Wu *et al*. [15], it reduces 11.7% and it is significant.

V. CONCLUSIONS

Cluster tools are widely used to process wafers in semiconductor manufacturing. Recently, diversified customer needs as well as corrective and preventive maintenance lead to frequent close-down and start-up processes for them. For some wafer manufacturing processes characterized by wafer residency time constraint, which requires a processed wafer has to be removed from a chamber in a limited time, the scheduling problem is more challenging. This work represents the first one that studies the challenging problem of scheduling start-up process for time-constrained single-arm cluster tools with parallel chambers. Based on a generalized backward strategy, this work first develops a Petri net model to reveal the behavior during the start-up process, with which a linear program is formulated to search an optimal and feasible schedule, as a result, such a schedule 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.

ACKNOWLEDGEMENT

The financial support from the National Research Foundation of Singapore Delta-NTU Corporate Lab Program with the project reference of DELTA-NTU CORP-SMA-RP2 is gratefully acknowledged.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Dae-Kyu Kim, Tae-Eog Lee, and Hyun-Jung 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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, pp. 170-178, 2003.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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, pp. 267-280, 2010.

[18] 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.

[19] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou. Optimal One-Wafer Cyclic Scheduling of Time-Constrained Hybrid Multicluster via Petri Nets. *IEEE Transactions on Systems, Man and Cybernetics: Systems*. Online, DOI: 10.1109/TSMC.2016.2531697, 2017.

[20] 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, 2018.

[21] 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.

[22] 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, DOI: 10.1109/TSMC.2016.2598303, 2017.

[23] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Petri Net-Based Optimal One-Wafer Cyclic Scheduling of Hybrid Multi-Cluster Tools in Wafer Fabrication," *IEEE Transactions on Semiconductor Manufacturing*, vol. 27, no. 2, pp. 192-203, 2014.

[24] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Optimal One-Wafer Cyclic Scheduling of Single-Arm Multi-Cluster Tools with Two-Space Buffering Modules," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, vol. 44, no. 12, pp. 1584-1597, 2014.

[25] F. J. Yang, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Petri Net-Based Polynomially Complex Approach to Optimal One-Wafer Cyclic Scheduling of Hybrid Multi-Cluster Tools in Semiconductor Manufacturing," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, vol. 44, no. 12, pp. 1598-1610, 2014.

[26] F. J. Yang, N. Q. Wu, Y. Qiao, and 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.