# Post-Processing Time-Aware Optimal Scheduling of Single Robotic Cluster Tools

QingHua Zhu, *Senior Member, IEEE*, Yan Qiao, *Member, IEEE*, NaiQi Wu, *Fellow, IEEE*, and Yan Hou

***Abstract*—Integrated circuit chips are produced on silicon wafers. Robotic cluster tools are widely used since they provide a reconfigurable and efficient environment for most wafer fabrication processes. Recent advances in new semiconductor materials bring about new functionality for integrated circuits. After a wafer is processed in a processing chamber, the wafer should be removed from there as fast as possible to guarantee its high-quality integrated circuits. Meanwhile, maximization of the throughput of robotic cluster tools is desired. This work aims to perform post-processing time-aware scheduling for such tools subject to wafer residency time constraints. To do so, closed-form expression algorithms are derived to compute robot waiting time accurately upon the analysis of particular events of robot waiting for single-arm cluster tools. Examples are given to show the application and effectiveness of the proposed algorithms.**

***Index Terms*—Cluster tool, discrete event systems, optimization, robotic systems, scheduling.**

## I. Introduction

To produce integrated circuit chips, a silicon wafer goes through a great number of fabrication procedures, up to hundreds of steps. Many of these wafer fabrication steps are performed using cluster tools [1], [2]. Typically, four to six processing machines/modules (PM) radially surround a robot to form a cluster tool in a vacuum environment, as illustrated in Fig. 1. The loadlock cassette modules (LL) are used to import/export raw/processed wafers. The robot in the center is responsible for transferring wafers between PMs/LLs. Depending on the number of blades (one or two), a single or dual-arm robot can handle one or two wafers at a time, respectively. The robot unloads a raw wafer from LLs, transfers it to PMs for processing pursuant to a predefined recipe, and returns the completed wafer to LLs [2], [3].

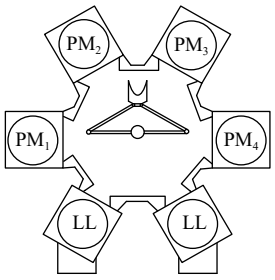



Fig. 1. A single-arm cluster tool.

Cluster tools can perform most wafer fabrication processes, including etching, vapor deposition, wafer cleaning and so on, making them increasingly prevalent in the fabrication processes. Just as done for scheduling complex production systems [4]–[7], much effort has been made on modeling, analysis and scheduling of cluster tools [8]–[24]. Chemical vapor deposition (CVD) used in various wafer fabrication processes require the avoidance of excessive exposure to mixed chemical gases at high temperatures. A wafer thus must be unloaded within a short time from a processing chamber after its processing is completed.

Abundant work has been done for scheduling cluster tools that are subject to such wafer residency time constraints [3], [25]–[29]. In particular, Kim *et al*. [30], Lee and Park [31], and Zuberek [29] have investigated the optimal scheduling problems of dual-arm cluster tools subject to wafer residency time constraints. For cluster tools under such constraints, further work is done in [32]–[35] and analytical-expression-based algorithms are proposed to find periodic optimal schedules whenever a feasible schedule exists. Petri nets have been effectively applied to model cluster tools [32]–[36] and other discrete event systems [37], [38].

A cluster tool starts its operation via a start-up transient process when its robot unloads the first wafer from LLs [39]. Then, it enters the steady state [40], and eventually it undergoes a close-down transient process to terminate its operation when no raw wafers are released from LLs. Yi *et al*. [40] handle the operations under the steady state. Under the steady state, cycle time is referred to as the time taken for finishing a wafer in a repetitive manner [1]. In recent years, due to preferences for small lot production [14], [41], [42] and maintenance demands [43], [44], transient periods including start-up and close-down processes increase to a large proportion during the whole wafer fabrication. The work in

Manuscript received April 2, 2019; revised May 11, 2019; accepted June 11, 2019. This work was supported in part by the National Natural Science Foundation of China (61673123 , 61803397, 61603100) , Science and Technology Development Fund (FDCT), and Macau SAR of China (0017/2019/Al, 005/2018/Al, 011/2017/A). Recommended by Associate Editor MengChu Zhou. *(Corresponding author: NaiQi Wu.)*

Citation: Q. H. Zhu, Y. Qiao, N. Q. Wu, and Y. Hou, "Post-processing time-aware optimal scheduling of single robotic cluster tools," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 597–605, Mar. 2020.

Q. H. Zhu and Y. Hou are with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China (e-mail: zhuqh@gdut.edu.cn; houyan@gdut.edu.cn).

Y. Qiao is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa 999078, Macau, China (e-mail: yqiao@must.edu.mo).

N. Q. Wu is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa 999078, Macau, China and also with the State Key Laboratory of Precision Electronic Manufacturing Technology and Equipment, Guangdong University of Technology, Guangzhou 510006, China (e-mail: nqwu@must.edu.mo).

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

Digital Object Identifier 10.1109/JAS.2020.1003069

[45] studies a generalized backward sequence and workload-based conditions to minimize makespan for scheduling such transient processes of single-arm cluster tools with parallel PMs. Kim *et al*. [46] developed latest and earliest starting policies to minimize start-up and close-down periods for dual-arm cluster tools under wafer residency time constraints. For start-up and close-down periods of single-arm cluster tools with wafer residency time constraints, the work in [47]–[49] analyzes schedulability conditions and finds optimal schedules. Kim *et al*. [50] investigate the scheduling problem of start-up and close-down periods for cluster tools subject to task time variation and wafer residency time constraints.

Thanks to significant advancements of semiconductor nanomaterial [51], such as carbon nanotubes and graphene, up-to-date wafer circuit line width has been reduced to less than ten nanometers. As an example, CVD involves a chemical reaction between a mixture of gasses and a wafer's surface that takes place at temperatures up to 1000ºC. For thin line width circuits, there are complex and delicately controllable growth kinetic and reactions to form the circuit layer (s) through CVD. The duration of certain sub-steps has a great influence on the high-quality synthesis of monolayer, bilayer, or few-layer graphene [52]. The quality of devices fabricated on a wafer is heavily dependent on reaction time [51], [53]. Thus, there is an upper-limit with regards to time for which a wafer may stay in a chamber after its processing. If the wafer remains in the chamber past this time limit, it would typically be scrapped. It is crucial to minimize post-processing time for large wafers so that their surfaces absorb fewer by-products in the chamber and can be fabricated uniformly [52]. Therefore, it is desirable to maximize the productivity of wafer fabrication, while minimizing the post-processing time to satisfy the requirement for yielding high-quality circuits.

To the authors' best knowledge, this issue has not been tackled in the literature yet. The main difference between this work and the prior work on scheduling cluster tools is that the latter [3], [10], [13], [54], including our previous work [32], [49], does not consider post-processing time minimization. Since post-processing time minimization and throughput maximization may be in conflict, it can be extremely challenging to find an optimal schedule to optimize both. Furthermore, determining a one-wafer schedule which is simple and easy to understand and implement by practitioners is not obvious. The semiconductor industry prefers one-wafer cyclic scheduling during which after a sequence of robot actions is performed, a cluster tool maintains the exact same state throughout the entire process [10]. Owing to the high capital cost of cluster tools, maximizing throughout is of significant importance in wafer fabrication, we aim to achieve minimization of post-processing time. Despite the scheduling algorithms in prior work which guarantees that post-processing time does not exceed its upper-limit, it is not advised that the post-processing time of a wafer is longer at some steps while it is shorter at other steps because of quality considerations.

This work has twofold contributions: 1) an optimal one-wafer cyclic schedule that maximizes a cluster tool's throughput is found and then the post-processing time is minimized for single-arm cluster tools; and 2) the post-processing time difference among the processing steps is minimized as much as possible. This work proposes algorithms to get an exactly optimal solution in linear-time method instead of a population-based optimization one [55]–[58].

The remainder of this paper is structured as follows. Section II presents the fundamental temporal properties of scheduling single-arm cluster tools. Section III presents algorithms to find a schedule that can minimize cycle time and total post-processing time for single-arm cluster tools. Section IV demonstrates the application of the proposed algorithms. Section V summarizes this work.

## II. Temporal Properties of Single-arm Cluster Tools

With $n$ processing steps in a cluster tool, let $\mathbb{N}_n = \{1, 2, \ldots, n\}$ and $\Omega_n = \{0\} \cup \mathbb{N}_n$. For easy presentation, suppose that each processing step is configured with one PM. For the example shown in Fig. 1, without loss of generality, it is assumed that the wafer processing route is $\langle \mathrm{LL} \rightarrow \mathrm{PM}_1 \rightarrow \mathrm{PM}_2 \rightarrow \mathrm{PM}_3 \rightarrow \mathrm{PM}_4 \rightarrow \mathrm{LL} \rangle$. For single-arm cluster tools, let $\lambda$ and $\mu$ denote the time taken for robot loading/unloading a wafer into/from LLs, and robot moving between any two PMs or between a PM and an LL, respectively. Let $\omega_i$ denote robot waiting time before unloading a wafer from $\mathrm{PM}_i$. If the workload bottleneck is the robot, the cluster tool runs in a transport-bound mode; while if the bottleneck is a PM, it runs in a process-bound mode. Backward scheduling is optimal for the steady state of single-arm cluster tools operating in a process-bound mode [10]. By this scheduling rule, the robot performs the operation sequence as follows:

$\langle$ moving to $\mathrm{PM}_n \rightarrow$ waiting there for $\omega_n$ time units $\rightarrow$ unloading a wafer from $\mathrm{PM}_n \rightarrow$ moving to LLs $\rightarrow$ loading the wafer into LLs $\rightarrow$ moving to $\mathrm{PM}_{n-1} \rightarrow$ waiting there for $\omega_{n-1}$ time units $\rightarrow$ unloading a wafer from $\mathrm{PM}_{n-1} \rightarrow$ moving to $\mathrm{PM}_n \rightarrow$ loading the wafer into $\mathrm{PM}_n \rightarrow$ moving to $\mathrm{PM}_{n-2} \rightarrow$ waiting there for $\omega_{n-2}$ time units $\rightarrow$ unloading a wafer from $\mathrm{PM}_{n-2} \rightarrow \ldots \rightarrow$ unloading a wafer from $\mathrm{PM}_1 \rightarrow \ldots \rightarrow$ moving to LLs $\rightarrow$ waiting there for $\omega_0$ time units $\rightarrow$ unloading a wafer from LLs $\rightarrow$ moving to $\mathrm{PM}_1 \rightarrow$ loading the wafer into $\mathrm{PM}_1 \rightarrow$ moving to $\mathrm{PM}_n$ again$\rangle$.

Let $\alpha_i$ denote the wafer processing time at Step $i$ or $\mathrm{PM}_i$. After a wafer is completed at Step $i$, it cannot stay there for more than $\delta_i$ ($\geq 0$) time units, which is also the upper limit of post-processing time at Step $i$. Some temporal properties of single-arm cluster tools are recalled as follows [32].

The robot cycle time is

$$\psi = 2(n+1)(\lambda+\mu) + \sum_{i=0}^{n} \omega_i = \psi_1 + \psi_2 \tag{1}$$

where the robot's task time $\psi_1 = 2(n+1)(\lambda+\mu)$ is a constant, while $\psi_2 = \sum_{i=0}^{n} \omega_i$ is its waiting time in a cycle.

At Step $i$, the lower bound time required to finish a wafer is

$$\Pi_{iL} = \alpha_i + 4\lambda + 3\mu, \quad i \in \mathbb{N}_n \tag{2}$$

and the upper bound time required to finish a wafer is

$$\Pi_{iU} = \alpha_i + 4\lambda + 3\mu + \delta_i, \quad i \in \mathbb{N}_n. \tag{3}$$

Let $\tau_i$ denote the wafer sojourn time at Step $i$. A wafer should stay in $PM_i$ for $\tau_i \geq \alpha_i$ time to complete its processing, which can be calculated as

$$\tau_i = \psi - (4\lambda + 3\mu + \omega_{i-1}), \quad i \in \mathbb{N}_n. \tag{4}$$

## III. Post-processing Time-aware Scheduling

Let $r_i = \tau_i - \alpha_i$ denote the post-processing time at Step $i$. By (4), it is known that if $\omega_{i-1}$ ($i \in \mathbb{N}_n$) increases, then $r_i$ decreases. The key issue is to minimize the total post-processing time $\sum_{i=1}^{n} r_i$.

For a single-arm cluster tool in a process-bound mode, its lower bound of cycle time is $\Pi = \max\{\Pi_{iL}, i \in \mathbb{N}_n\}$. When a cluster tool operates in a one-wafer cyclic schedule, the robot and all the processing steps operate in a paced way, i.e., $\psi = \Pi$. Our goal is to schedule the robot to achieve its cycle time as $\Pi$ so as to minimize the cycle time for a tool. To avoid violating wafer residency time constraints, schedulability conditions must be established. Upon these conditions, schedules are found to optimize these two objectives. In (1), we have $\psi_2 = \sum_{i=0}^{n} \omega_i$, which implies that if a feasible schedule exists, we must properly assign $\psi_2$ into $\omega_i$, $i \in \Omega_n$, such that the obtained schedule is feasible. When the cycle time is minimal and wafer residency time constraints are satisfied, $\sum_{i=1}^{n} r_i$ must be minimized in order to guarantee high-quality circuits on a wafer. Furthermore, when $\sum_{i=1}^{n} r_i$ is minimized, one must have a uniform substrate across the wafer, or the post-processing time should be evenly distributed among the processing steps. Note that there may be a variety of feasible schedules by assigning $\psi_2$ into $\omega_i$, $i \in \Omega_n$. The optimal one achieves the uniformity among $r_i$'s or $r_i = r_j$, $i, j \in \mathbb{N}_n$. To obtain such a result, we minimize the sum of post-processing time $r_i$'s, which is one of the objectives. In this work, since $\lambda$, $\mu$, and $\alpha_i$'s ($i \in \mathbb{N}_n$) are constants, if $\omega_i$'s are determined, a schedule is found as well.

Based on the above discussion, we propose Algorithm 1 to assign $\psi_2$ into $\omega_i$, $i \in \Omega_n$, so as to obtain a schedule. To calculate a schedule, we must ensure that a feasible schedule exists, since otherwise, it is meaningless. There are two cases where a feasible schedule exists.

*Case 1:* One of the following conditions should be met: 1) $\Pi \leq \Pi_{iU}$ and $\psi_1 \leq \Pi$, $i \in \mathbb{N}_n$; and 2) $\Pi_{iL} \leq \psi_1 \leq \Pi_{iU}$, $i \in \mathbb{N}_n$. Note that $|V|$ denotes the cardinality of set $V$. Condition 1) means that the workloads of all processing step are relatively balanced and the cluster tool is process-bound, while Condition 2 says that the robot is always busy, i.e., the cluster tool is transport-bound.

**Algorithm 1.** Scheduling single-arm cluster tools for Case 1

Input: $\lambda, \mu, \alpha_i, \delta_i$ ($i \in \mathbb{N}_n$)
Output: $\omega_i$ ($i \in \Omega_n$)
1. $\psi_1 \leftarrow 2(n+1)(\lambda + \mu)$
2. $\Pi_{iL} \leftarrow \alpha_i + 4\lambda + 3\mu$ and $\Pi_{iU} \leftarrow \Pi_{iL} + \delta_i$, $i \in \mathbb{N}_n$
3. $\Pi \leftarrow \max\{\Pi_{iL}, i \in \mathbb{N}_n\}$
4. **If** $\Pi \leq \Pi_{iU}$ and $\psi_1 \leq \Pi$, $i \in \mathbb{N}_n$ **Then**
5. $\omega_0 \leftarrow \min\{\Pi - (\alpha_1 + 4\lambda + 3\mu), \Pi - \psi_1\}$
6. $\omega_{i-1} \leftarrow \min\{\Pi - (\alpha_i + 4\lambda + 3\mu), \Pi - \psi_1 - \sum_{k \in \Omega_{i-1} \setminus \{i-1\}} \omega_k\}$ for $i \in \mathbb{N}_n \setminus \{1\}$
7. $\omega_n \leftarrow \Pi - \psi_1 - \sum_{i=0}^{n-1} \omega_i$
8. $r_i \leftarrow \Pi - (\alpha_i + 4\lambda + 3\mu + \omega_{i-1})$ for $i \in \mathbb{N}_n$
9. $r_0 \leftarrow \Pi - (4\lambda + 3\mu + \omega_n)$
10. $V \leftarrow \{i |$ ($\Pi_{iL} < \Pi$ and $\omega_{i-1} > 0$) or ($r_i > 0$ and $\omega_{i-1} = 0$), $i \in \mathbb{N}_n\}$
11. **If** ($\omega_n > 0$ and $r_0 = 0$) **Then** $V \leftarrow V \cup \{0\}$ **EndIf**
12. $\Delta \leftarrow \sum_{i \in V} r_i / |V|$
13. **If** $\Delta \leq \Pi - (\alpha_i + 4\lambda + 3\mu)$ and $i \in V$ **Then**
14. **For** each $i \in V$ **do**
15. $r_i \leftarrow \Delta$
16. **If** $i \neq 0$ **Then** $\omega_{i-1} \leftarrow \Pi - (\alpha_i + 4\lambda + 3\mu + r_i)$
17. **Else** $\omega_0 \leftarrow \Pi - (\alpha_1 + 4\lambda + 3\mu + r_1)$
18. **EndIf**
19. **EndFor**
20. **EndIf**
21. **EndIf**
22. **If** $\Pi_{iL} \leq \psi_1 \leq \Pi_{iU}$ ($i \in \mathbb{N}_n$) **Then**
23. $\omega_i \leftarrow 0$, for $i \in \Omega_n$
24. **EndIf**

The following result is given to show the optimality of the obtained schedule via Algorithm 1.

*Theorem 1:* For single-arm cluster tools subject to wafer residency time constraints, if one of the two conditions: 1) $\Pi \leq \Pi_{iU}$ and $\psi_1 \leq \Pi$, $i \in \mathbb{N}_n$; and 2) $\Pi_{iL} \leq \psi_1 \leq \Pi_{iU}$, $i \in \mathbb{N}_n$, is satisfied, Algorithm 1 finds a schedule to reach the lower bound of cycle time and minimize total post-processing time.

*Proof:* Set the lower bound $\Pi$ as the cycle time of a cluster tool $\psi = \Pi$, or the minimal cycle time. Next, we check: a) whether a feasible schedule with cycle time $\Pi$ can be constructed or not to meet the residency constraints under one of the given conditions; and b) minimize the post-processing time $\sum_{i=1}^{n} r_i$.

1) Conditions $\Pi \leq \Pi_{iU}$ and $\psi_1 \leq \Pi$, $i \in \mathbb{N}_n$, hold. Since $\tau_i = \Pi - (4\lambda + 3\mu + \omega_{i-1})$, $i \in \mathbb{N}_n$, implying that $\tau_i$ decreases if $\omega_{i-1}$ increases, we need to maximize $\omega_i$'s ($i \in \Omega_n \setminus \{n\}$) so that $\tau_i$ ($i \in \mathbb{N}_n$) can be minimized.

By Line 5 of Algorithm 1, if $\omega_0 = \Pi - (\alpha_1 + 4\lambda + 3\mu) < \Pi - \psi_1$, then $\tau_1 = \Pi - (4\lambda + 3\mu + \omega_0) = \alpha_1$ is minimized and $\tau_1 \in [\alpha_1, \alpha_1 + \delta_1]$. If $\omega_0 = \Pi - \psi_1 < \Pi - (\alpha_1 + 4\lambda + 3\mu)$, $\omega_0$ is maximized because no more time than $\Pi - \psi_1$ can be assigned to $\omega_0$. Then, $\tau_1$ is minimized and $\tau_1 = \Pi - (4\lambda + 3\mu + \omega_0) = \Pi - (4\lambda + 3\mu) - (\Pi - \psi_1) \geq \Pi - (4\lambda + 3\mu) - (\Pi - (\alpha_1 + 4\lambda + 3\mu)) = \alpha_1$, $\tau_1 = \Pi - (4\lambda + 3\mu + \omega_0) \leq \Pi_{1U} - (4\lambda + 3\mu + \omega_0) \leq \Pi_{1U} - (4\lambda + 3\mu) = \alpha_1 + \delta_1$.

By Line 6 of Algorithm 1, if $\omega_{i-1} = \Pi - (\alpha_i + 4\lambda + 3\mu) < \Pi - \psi_1$, then $\tau_i = \Pi - (4\lambda + 3\mu + \omega_{i-1}) = \alpha_i$ is minimized and $\tau_i \in [\alpha_i, \alpha_i + \delta_i]$. If $\omega_{i-1} = \Pi - \psi_1 - \sum_{k \in \Omega_{i-1} \setminus \{i-1\}} \omega_k \leq \Pi - (\alpha_i + 4\lambda + 3\mu)$, $\omega_{i-1}$ is maximized because no more time than $\Pi - \psi_1 - \sum_{k \in \Omega_{i-1} \setminus \{i-1\}} \omega_k$ can be assigned to $\omega_{i-1}$. Then, $\tau_i$ is minimized and $\tau_i = \Pi - (4\lambda + 3\mu + \omega_{i-1}) \geq \Pi - (4\lambda + 3\mu + (\Pi - (\alpha_i + 4\lambda + 3\mu))) = \alpha_i$, and $\tau_i = \Pi - (4\lambda + 3\mu + \omega_{i-1}) \leq \Pi_{iU} - (4\lambda + 3\mu + \omega_{i-1}) = \alpha_i + \delta_i - \omega_{i-1} \leq \alpha_i + \delta_i$.

Thus, wafer residency constraints are satisfied when $\omega_i$ ($i \in \Omega_n$) is set by Lines 5–7 of Algorithm 1 while $\sum_{i=1}^{n} r_i$ is minimized. Then, Lines 8–20 intend to readjust the post-processing time evenly for Steps in $V = \{i | (\Pi_{iL} < \Pi$ and $\omega_{i-1} > 0)$ or $(r_i > 0$ and $\omega_{i-1} = 0)$, $i \in \mathbb{N}_n\}$. Furthermore, during the

adjustment process, $\sum_{i=1}^{n} r_i$ keeps unchanged.

If $\Delta \le \Pi - (\alpha_i + 4\lambda + 3\mu)$ ($i \in V$) holds in Line 13, then, by Lines 16 and 17, we have $\omega_{i-1} = \Pi - (\alpha_i + 4\lambda + 3\mu + \Delta) \ge \Pi - (\alpha_i + 4\lambda + 3\mu + \Pi - (\alpha_i + 4\lambda + 3\mu)) = 0$ ($i \ne 0$) and $\omega_0 = \Pi - (\alpha_1 + 4\lambda + 3\mu + \Delta) \ge \Pi - (\alpha_i + 4\lambda + 3\mu + \Pi - (\alpha_i + 4\lambda + 3\mu)) = 0$. Obviously, for $i \in V$, $r_i = \Delta > 0$ and $r_i = \Delta \le \Pi - (\alpha_i + 4\lambda + 3\mu) \le \Pi_{iU} - (\alpha_i + 4\lambda + 3\mu) = \delta_i$.

2) Condition $\Pi_{iL} \le \psi_1 \le \Pi_{iU}$ ($i \in \mathbb{N}_n$) holds as shown in Line 22 in Algorithm 1, we have $\omega_i = 0$, $i \in \Omega_n$. Then, $\tau_i = \Pi - (4\lambda + 3\mu + \omega_{i-1}) = \Pi - (4\lambda + 3\mu) \ge \Pi_{iL} - (4\lambda + 3\mu) = \alpha_i$, $\tau_i = \Pi - (4\lambda + 3\mu) \le \Pi_{iU} - (4\lambda + 3\mu) = \alpha_i + \delta_i$.

So far, we can conclude that if one of the given conditions in this theorem is satisfied, Algorithm 1 can find a schedule that minimizes both the cycle time and wafer post-processing time. ■

By Lines 5–7, Algorithm 1 initially finds a schedule with both cycle time and wafer post-processing time being minimized. Lines 8–20 make efforts to readjust post-processing time evenly for some steps. This adjustment can avoid unnecessarily excessive post-processing time, which is beneficial in improving the quality of the fabricated wafer.

In Algorithm 1, all the statements make calculations based on closed-form expressions. The number of iterations in the For-loop of Lines 14–19 cannot be greater than $n$. It is obvious that the computational complexity of Algorithm 1 is $O(n)$.

*Case 2:* $\bigcap_{k\in\mathbb{N}_n} [\Pi_{kL}, \Pi_{kU}] = \emptyset$, which implies that the workloads of the processing steps are unbalanced. In this case, Algorithm 2 is proposed to determine whether a feasible schedule can be found, and whether the lower bound of cycle time of the found schedule is reached.

**Algorithm 2.** Scheduling single-arm cluster tools for Case 2

Input: $\lambda, \mu, \alpha_i, \delta_i$ ($i \in \mathbb{N}_n$)
Output: $\Gamma, \omega_i$ ($i \in \Omega_n$)
1. $\Gamma \leftarrow$ True, $\psi_1 \leftarrow 2(n+1)(\lambda + \mu)$
2. $\Pi_{iL} \leftarrow \alpha_i + 4\lambda + 3\mu$, $\Pi_{iU} \leftarrow \Pi_{iL} + \delta_i$, $i \in \mathbb{N}_n$
3. $\Pi \leftarrow \max\{\Pi_{iL}, i \in \mathbb{N}_n\}$
4. $E \leftarrow \{k | \Pi_{kU} < \Pi, k \in \mathbb{N}_n\}$, $F \leftarrow \mathbb{N}_n \setminus E$
5. $\omega_{i-1} \leftarrow \Pi - (\alpha_i + \delta_i + 4\lambda + 3\mu)$, for $i \in E$
6. $\omega_{i-1} \leftarrow 0$, for $i \in F$
7. **If** $\sum_{i\in E} \omega_{i-1} > \Pi - \psi_1$ **Then**
8. $\Gamma \leftarrow$ False, return //Unschedulable.
9. **EndIf**
10. $G \leftarrow \{i | i \in F$ and $\Pi_{iL} < \Pi\}$
11. $\omega_{i-1} \leftarrow 0$, $i \in F \setminus G$
12. $\psi \leftarrow \Pi - \psi_1 - \sum_{i\in E} \omega_{i-1}$
13. $h \leftarrow \sum_{i\in G} (\Pi - (\alpha_i + 4\lambda + 3\mu))$
14. **If** $\psi > h$ **Then**
15. $\omega_{i-1} \leftarrow \Pi - (\alpha_i + 4\lambda + 3\mu)$, $i \in G$
16. $\omega_n \leftarrow \psi - h$
17. $r_i \leftarrow \Pi - (\alpha_i + 4\lambda + 3\mu + \omega_{i-1})$ for $i \in \mathbb{N}_n$
18. **Else**
19. $\Upsilon \leftarrow h - \psi$, $H \leftarrow G$
20. $\omega_n \leftarrow 0$
21. **Do**
22. $A \leftarrow \{i | i \in H$ and $\Upsilon/|H| > \Pi - (\alpha_i + 4\lambda + 3\mu)\}$
23. **If** $A \ne \emptyset$ **Then**
24. $r_i \leftarrow \Pi - (\alpha_i + 4\lambda + 3\mu)$, $i \in A$
25. $\omega_{i-1} \leftarrow \Pi - (\alpha_i + 4\lambda + 3\mu + r_i)$, $i \in A$
26. $\Upsilon \leftarrow \Upsilon - \sum_{i\in A} r_i$, $H \leftarrow H \setminus A$
27. **If** $\Upsilon < 0$ **Then** goto Adjust **EndIf**
28. **Else**
29. $r_i \leftarrow \Upsilon/|H|$, $i \in H$
30. $\omega_{i-1} \leftarrow \Pi - (\alpha_i + 4\lambda + 3\mu + r_i)$, $i \in H$
31. $\Upsilon \leftarrow \Upsilon - \sum_{i\in H} r_i$
32. $H \leftarrow \emptyset$
33. **EndIf**
34. **While** $H \ne \emptyset$
35. **If** $\Upsilon < 0$ **Then**
36. **Adjust:** $\psi_G \leftarrow 0$
37. **For** $i \in G$
38. $\omega_{i-1} \leftarrow \min\{\Pi - (\alpha_i + 4\lambda + 3\mu), \psi - \psi_G\}$
39. $r_i \leftarrow \Pi - (\alpha_i + 4\lambda + 3\mu + \omega_{i-1})$
40. $\psi_G \leftarrow \psi_G + \omega_{i-1}$
41. **EndFor**
42. **EndIf**
43. **EndIf**

*Theorem 2:* If $\bigcap_{k\in\mathbb{N}_n} [\Pi_{kL}, \Pi_{kU}] = \emptyset$, Algorithm 2 can determine whether a feasible schedule exists. If it does, Algorithm 2 can find a schedule that minimizes both the cycle time and post-processing time.

*Proof:* By Line 4 of Algorithm 2, we have $E = \{k | \Pi_{kU} < \Pi, k \in \mathbb{N}_n\}$. For Steps $i \in E$, since the cycle time of every step must be identical to obtain a one-wafer cyclic schedule, $\omega_i$'s ($i \in E$) must be set by Line 5, which means that $\omega_i$'s ($i \in E$) have been maximized. Thus, by $\tau_i = \Pi - (4\lambda + 3\mu + \omega_{i-1})$, $\tau_i$ 's ($i \in E$) are minimized.

If $\sum_{i\in E} \omega_{i-1} \le \Pi - \psi_1$ holds, it is feasible to find a schedule by Lines 5 and 6. If not, there is no feasible schedule as specified by Lines 7–9.

Lines 4 and 10 define that $F = \mathbb{N}_n \setminus E$ and $G = \{i | i \in F$ and $\Pi_{iL} < \Pi\}$. For Steps $i \in F \setminus G$, we have $\Pi_{iL} = \Pi$. By Line 11, $\omega_{i-1} = 0$, then $\tau_i = \Pi - (4\lambda + 3\mu + \omega_{i-1}) = \Pi - (4\lambda + 3\mu) = \Pi_{iL} - (4\lambda + 3\mu) = \alpha_i \in [\alpha_i, \alpha_i + \delta_i]$ and $r_i = \tau_i - \alpha_i = 0$ is minimized.

Next, check if $r_i$ ($i \in G$) is minimized and the residency time constraints for Steps 1–$n$ are satisfied. Lines 12 and 13 set $h = \sum_{i\in G} (\Pi - (\alpha_i + 4\lambda + 3\mu))$ and $\psi = \Pi - \psi_1 - \sum_{i\in E} \omega_{i-1}$. There are two cases which are discussed below.

*Case A:* $\psi > h$.

By Line 15, for Steps $i \in G$, $\omega_{i-1} = \Pi - (\alpha_i + 4\lambda + 3\mu)$, then $\tau_i = \Pi - (4\lambda + 3\mu + \omega_{i-1}) = \alpha_i \in [\alpha_i, \alpha_i + \delta_i]$ and $r_i = \tau_i - \alpha_i = 0$ is minimized. Hence, $r_i$ is minimized for Steps $i \in F$.

*Case B:* $\psi \le h$.

Lines 4, 12, 20, and 22 of Algorithm 2 present that $A = \{i | i \in H$ and $\Upsilon/|H| > \Pi - (\alpha_i + 4\lambda + 3\mu)\}$, implying that $\Pi_{iL} < \Pi \le \Pi_{iU}$ and $\Upsilon/|H| > \Pi - (\alpha_i + 4\lambda + 3\mu)$.

If $A \ne \emptyset$, i.e., $\Upsilon/|H| > \Pi - (\alpha_i + 4\lambda + 3\mu)$, then by Lines 24 and 25, we have $\omega_{i-1} = \Pi - (\alpha_i + 4\lambda + 3\mu + \Pi - (\alpha_i + 4\lambda + 3\mu)) = 0$ for Steps $i \in A$. Thus, by Line 24, $r_i = \Pi - (\alpha_i + 4\lambda + 3\mu) \ge \Pi_{iL} - (\alpha_i + 4\lambda + 3\mu) = 0$, $r_i \le \Pi_{iU} - (\alpha_i + 4\lambda + 3\mu) = \delta_i$. By Line 27, if $\Upsilon > 0$, i.e., there are $\Upsilon$ time units that can be assigned to the post-processing time at the steps in $H$. If $\Upsilon < 0$ holds, then it leads to a contradictory as follows. If $\Upsilon < 0$, then $h - \psi - \sum_{i\in G\setminus H} r_i < 0$, or $\sum_{i\in G\setminus H} r_i > h - \psi$. If Line 27 is removed and all $\omega_{i-1}$'s ($i \in G$) are set by this "do-while" loop in Lines 21–34, then $\sum_{i\in G} \omega_{i-1} = \sum_{i\in G} \Pi - (\alpha_i + 4\lambda + 3\mu + r_i) = \sum_{i\in G} (\Pi - (\alpha_i +$

$4\lambda+3\mu)) - \sum_{i\in G} r_i = h - \sum_{i\in G} r_i$. Equation (1) and Lines 11, 12 and 20 imply $\sum_{i\in G} \omega_{i-1} = \psi$. Thus, $\sum_{i\in G} r_i = h - \psi$, which is contradictory to the aforementioned $\sum_{i\in G\setminus H} r_i > h - \psi$ because of $\sum_{i\in \mathrm{H}} r_i > 0$. Thereby, if $\Upsilon < 0$ occurs in Line 27, Lines 36–41 can reset $\omega_{i-1}$'s ($i \in G$). By Line 38, if $\omega_{i-1} = \Pi - (\alpha_i + 4\lambda + 3\mu) < \psi - \psi_G$, then $r_i = \Pi - (\alpha_i + 4\lambda + 3\mu + \omega_{i-1}) = 0$ is minimized. By Line 38, if $\psi - \psi_{\mathrm{G}} < \Pi - (\alpha_i + 4\lambda + 3\mu)$, then there are $\psi - \psi_{\mathrm{G}}$ time units available to be assigned to $\omega_{i-1}$, or $\omega_{i-1} = \psi - \psi_{\mathrm{G}}$ is maximized. Then, $r_i = \Pi - (\alpha_i + 4\lambda + 3\mu + \omega_{i-1}) = \Pi - (\alpha_i + 4\lambda + 3\mu + \psi - \psi_G) > \Pi - (\alpha_i + 4\lambda + 3\mu + (\Pi - (\alpha_i + 4\lambda + 3\mu))) = 0$ and $r_i = \Pi - (\alpha_i + 4\lambda + 3\mu + \psi - \psi_G) \le \Pi - (\alpha_i + 4\lambda + 3\mu + 0) \le \Pi_{iU} - (\alpha_i + 4\lambda + 3\mu) = \delta_i$ because of $\psi - \psi_G \ge 0$.

If $A = \emptyset$ does not hold, then $\Upsilon/|H| \le \Pi - (\alpha_i + 4\lambda + 3\mu)$. This situation is handled by Lines 29–32, distributing post-processing time evenly. By Lines 29 and 30, for Steps $i \in H$, $\omega_{i-1} = \Pi - (\alpha_i + 4\lambda + 3\mu + \Upsilon/|H|)$, $r_i = \Upsilon/|H| \le \Pi - (\alpha_i + 4\lambda + 3\mu) \le \Pi_{iU} - (\alpha_i + 4\lambda + 3\mu) = \delta_i$. After Lines 29–32 are executed only once, $\Upsilon$ becomes zero and $H$ becomes $\emptyset$. This means that the condition $\Upsilon < 0$ in Line 35 cannot be met.

So far, for $i \in E \cup F$, we have $\sum_{i\in E\cup F} \omega_{i-1} = \sum_{i\in E} \omega_{i-1} + \sum_{i\in G} \omega_{i-1} + \sum_{i\in F\setminus G} \omega_{i-1} = \sum_{i\in E} \omega_{i-1} + (\Pi - \psi_1 - \sum_{i\in E} \omega_{i-1}) + 0 = \Pi - \psi_1$. Meanwhile, $\omega_n$ has been set to be zero by Line 20. ■

Algorithm 2 deals with the situation of $\bigcap_{k\in\mathbb{N}_n} [\Pi_{kL}, \Pi_{kU}] = \emptyset$ which indicates that some steps have heavier workloads than others. Given the lower bound of cycle time, robot waiting time $\omega_{i-1}$ ($i \in E$) is assigned in advance. After its assignment, one can decide if the cluster tool is schedulable as done in Line 7. Then we compare the workload of the robot with those of the processing steps, and if the robot is fast enough, the post-processing time can be minimized to be zero, which is implemented in Lines 15–17. If not, there is non-zero post-processing time for Steps $i \in G$. Then, the sum of them, i.e., $\sum_{i\in G} r_i$ is minimized and $r_i$ ($i \in E$) is set evenly, as implemented in Lines 19–42.

In Algorithm 2, all the statements make calculations based on closed-form expressions. The count of iterations in Lines 21–34 and 37–41 cannot be greater than $n$. It is obvious that the computational complexity of Algorithm 2 is $O(n)$.

## IV. Examples

In this section, we demonstrate how to apply the proposed method to find schedules for bi-objective problems. The time unit is seconds and is omitted thereafter. Let $(r'_1, r'_2, r'_3, r'_4)$ denote the post-processing time if a cluster tool is scheduled by a conventional method that does not consider minimizing the post-processing time.

*Example 1:* There are four steps in a single-arm cluster tool where every step is equipped with one PM. The robot waiting time is found by Algorithm 1. The activity time, wafer residency time constraints and the post-processing time at each step (excluding Step 0 or LLs) is given as follows:

$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4; \lambda, \mu) = (50, 66, 52, 50; 4, 2)$$
$$(\delta_1, \delta_2, \delta_3, \delta_4) = (20, 20, 20, 20)$$
$$(\omega_0, \omega_1, \omega_2, \omega_3, \omega_4) = (10, 0, 8, 10, 0)$$
$$(r_1, r_2, r_3, r_4) = (6, 0, 6, 6), \sum_{i=1}^{4} r_i = 18.$$

If the existing algorithms in [32] are applied to find an optimal schedule, we obtain post-processing time as $(r'_1, r'_2, r'_3, r'_4) = (16, 0, 14, 16)$ and $\sum_{i=1}^{4} r'_i = 46$. By our algorithm, the total post-processing time is decreased by 60.9%. Furthermore, $r_i$, $i \in \mathbb{N}_4$, is evenly distributed among the four steps.

The schedules obtained by Algorithm 1 and previous work in [32] are shown by Gantt charts in Fig. 2. The post-processing time is indicated by red bars in Figs. 2–4.

*Example 2:* There are four steps in a single-arm cluster tool where every step is equipped with one PM. It satisfies $\bigcap_{k\in\mathbb{N}_4} [\Pi_{kL}, \Pi_{kU}] = \emptyset$. The schedulability condition $\sum_{i\in E} \omega_{i-1} \le \Pi - \psi_1$ holds. Thus, the robot waiting time are found by Algorithm 2. The activity time, wafer residency time constraints, and the post-processing time at each step (excluding Step 0 or LLs) are

*Case 1:* $\psi > h$.

$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4; \lambda, \mu) = (85, 120, 110, 85; 5, 2)$$
$$(\delta_1, \delta_2, \delta_3, \delta_4) = (20, 20, 20, 20)$$
$$(\omega_0, \omega_1, \omega_2, \omega_3, \omega_4) = (15, 0, 10, 15, 36)$$
$$(r_1, r_2, r_3, r_4) = (20, 0, 0, 20), \sum_{i=1}^{4} r_i = 40.$$

If the algorithms in [32] are applied to find an optimal schedule, we have $(r'_1, r'_2, r'_3, r'_4) = (20, 0, 10, 20)$ and $\sum_{i=1}^{4} r'_i = 50$. The total post-processing time is decreased by 20%. The schedules obtained by Algorithm 2 and previous work in [32] are shown by Gantt charts in Fig. 3.

*Case 2:* $\psi < h$.

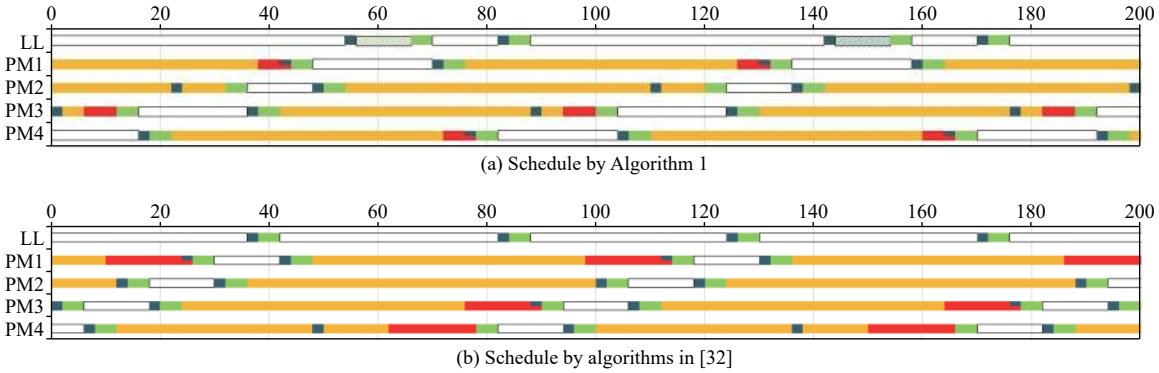
$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4; \lambda, \mu) = (36, 80, 78, 66; 4, 2)$$
$$(\delta_1, \delta_2, \delta_3, \delta_4) = (10, 10, 3, 14)$$
$$(\omega_0, \omega_1, \omega_2, \omega_3, \omega_4) = (34, 0, 0, 8, 0)$$
$$(r_1, r_2, r_3, r_4) = (10, 0, 2, 6), \sum_{i=1}^{4} r_i = 18.$$

The algorithms in [32] find an optimal schedule with $(r'_1, r'_2, r'_3, r'_4) = (10, 0, 2, 14)$ and $\sum_{i=1}^{4} r'_i = 26$. The total post-processing time is decreased by 30.8%. The schedules obtained by Algorithm 2 and previous work in [32] are shown by Gantt charts in Fig. 4.

We conclude that the proposed algorithms can find the same optimal-cycle-time schedule as the existing ones [32] but with significantly reduced post-processing time.

## V. Conclusions

Cluster tools are extensively adopted for wafer fabrication equipment in the semiconductor manufacturing industry. Wafers are fabricated in a complex chemical reaction environment where there are mixed gases and high-temperature heat. Characteristics of new materials and high-quality chips require that after their processing is completed, they should leave the processing chamber as soon as possible. The existing research ensures that post-processing time does not exceed the upper limit only. In order to obtain high-quality integrated circuits with advanced process control, this work considers two optimization objectives for single-arm cluster

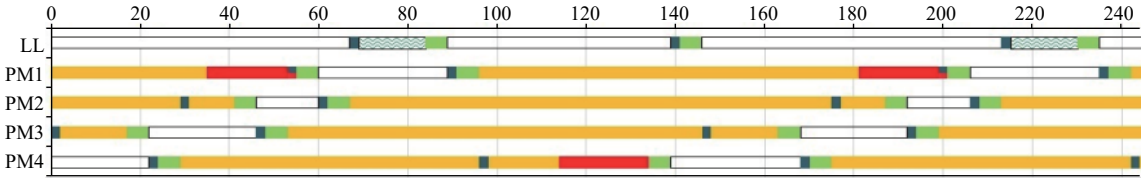



(a) Schedule by Algorithm 1

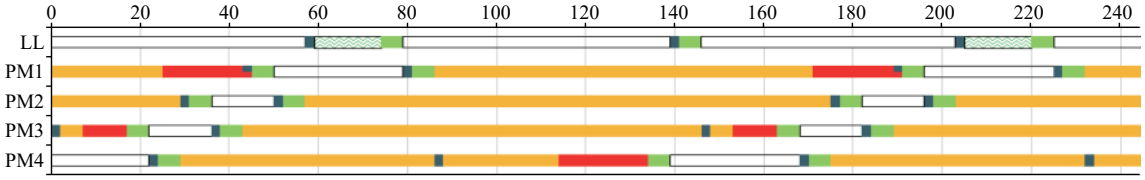



(b) Schedule by algorithms in [32]

Fig. 2. Gantt chart for Example 1.

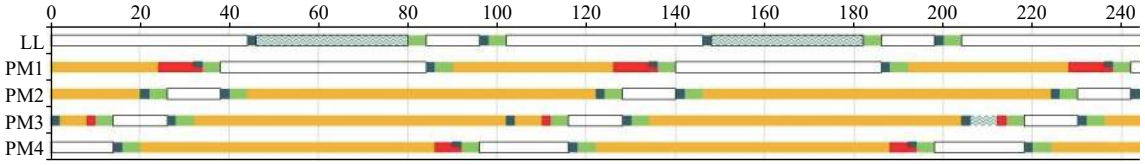



(a) Schedule by Algorithm 2

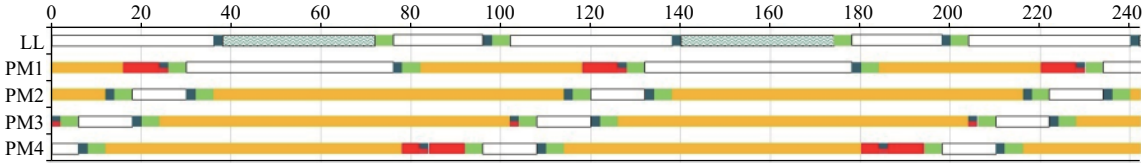



(b) Schedule by algorithms in [32]

Fig. 3. Gantt chart for Case 1 of Example 2.




(a) Schedule by Algorithm 2




(b) Schedule by algorithms in [32]

Fig. 4. Gantt chart for Case 2 of Example 2.

tools subject to wafer residency time constraints. It not only considers the maximization of the throughput but also minimizes wafer post-processing time in processing chambers. When the throughput is maximized and wafer residency time constraints are satisfied, we make efforts to shorten wafer sojourn time after a wafer is processed to avoid uneven post-processing time among the processing steps. Therefore, the obtained schedules are significantly better for fabricating high-quality wafers which are not seen in the existing reports to our best knowledge. In the future, we intend to answer how to schedule multi-cluster tools with multiple optimization goals.

## REFERENCES

[1] S. P. Sethi, J. B. Sidney, and C. Sriskandarajah, "Scheduling in dual gripper robotic cells for productivity gains," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 324–341, Jun. 2001.

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

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

[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 Trans. Syst., Man, Cybern.: Syst.*, vol. 47, no. 3, pp. 517–530, Mar. 2017.

[5] 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," *Inform. Sciences*, vol. 417, pp. 247–261, Nov. 2017.

[6] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Short-term scheduling of crude-oil operations: Petri net-based control-theoretic approach," *IEEE Robot. Autom. Mag.*, vol. 22, no. 2, pp. 64–76, Jun. 2015.

[7] 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," *Enterp. Inf. Syst.*, vol. 10, no. 6, May 2016.

[8] 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 Trans. Syst., Man, Cybern.: Syst.*, vol. 46, no. 10, pp. 1456–1467, Oct. 2016.

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

[10] M. Dawande, C. Sriskandarajah, and S. P. Sethi, "On throughput maximization in constant travel-time robotic cells," *Manufact. Serv. Oper. Manage*, vol. 4, no. 4, pp. 296–312, 2002.

[11] H. N. Geismar, M. Dawande, and S. Chelliah, "Robotic cells with parallel machines: throughput maximization in constant travel-time cells," *J. Scheduling*, vol. 7, no. 5, pp. 375–395, 2004.

[12] D. B. Jevtic, M. A. Pool, and R. S. Sunkara,"Method and apparatus for managing scheduling a multiple cluster tool," *European Patent*. vol. 1, 132, 792(A2) European: Applied Materials Inc., 2001.

[13] D.-K. Kim, H.-J. Kim, and T.-E. Lee, "Optimal scheduling for sequentially connected cluster tools with dual-armed robots and a single input and output module," *Int. J. Product. Res.*, vol. 55, no. 11, pp. 3092–3109, 2017.

[14] J.-H. Lee and H.-J. Kim, "Completion time analysis of wafer lots in single-armed cluster tools with parallel processing modules," *IEEE Trans. Automat. Sci. Eng.*, vol. 14, no. 4, pp. 1622–1633, 2017.

[15] X. Li and R. Y. K. Fung, "Optimal K-unit cycle scheduling of two-cluster tools with residency constraints and general robot moving times," *J. Scheduling*, vol. 19, no. 2, pp. 165–176, Apr. 2016.

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

[17] N. Q. Wu and M. C. Zhou, *System Modeling and Control With Resource-oriented Petri Nets*. Boca Raton, FL, USA: CRC Press, Inc., 2009.

[18] T. Nishi, Y. Watanabe, and M. Sakai, "An efficient deadlock prevention policy for noncyclic scheduling of multicluster tools," *IEEE Trans. Automat. Sci. Eng.*, vol. 15, no. 4, pp. 1677–1691, Oct. 2018.

[19] K. Park and J. R. Morrison, "Controlled wafer release in clustered photolithography tools: flexible flow line job release scheduling and an LMOLP heuristic," *IEEE Trans. Automat. Sci. Eng.*, vol. 12, no. 2, pp. 642–655, Apr. 2015.

[20] 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. Autom. Sinica*, vol. 5, no. 1, pp. 270–280, Mar. 2018.

[21] Q. H. Zhu, Y. Qiao, and N. Q. Wu, "Optimal integrated schedule of entire process of dual-blade multi-cluster tools from start-up to close-down," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 2, pp. 553–565, Mar. 2019.

[22] Q. H. Zhu, N. Q. Wu, Y. Qiao, and M. C. Zhou, "Scheduling of single-arm multi-cluster tools with wafer residency time constraints in semiconductor manufacturing," *IEEE Trans. Semiconduct. Manufact.*, vol. 28, no. 1, pp. 117–125, 2015.

[23] Q. H. Zhu, M. C. Zhou, Y. Qiao, and N. Q. Wu, "Scheduling transient processes for time-constrained single-arm robotic multi-cluster tools," *IEEE Trans. Semiconduct. Manufact.*, vol. 30, no. 3, pp. 261–269, Aug. 2017.

[24] H.-J. Kim and J.-H. Lee, "Closed-form expressions on lot completion time for dual-armed cluster tools with parallel processing modules," *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 2, pp. 898–907, Apr. 2019.

[25] C. Sriskandarajah, I. Drobouchevitch, and S. P. Sethi, "Scheduling multiple parts in a robotic cell served by a dual-gripper robot," *Oper. Res.*, vol. 52, no. 1, pp. 65–82, 2004.

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

[27] F. J. Yang, N. Q. Wu, Y. Qiao, M. C. Zhou, and Z. W. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.

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

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

[30] J.-H. Kim, T.-E. Lee, H.-Y. Lee, and D.-B. Park, "Scheduling analysis of time-constrained dual-armed cluster tools," *IEEE Trans. Semiconduct. Manufact.*, vol. 16, no. 8, pp. 521–534, Aug. 2003.

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

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

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

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

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

[36] J.-H. Paek and T.-E. Lee, "Optimal scheduling of dual-armed cluster tools without swap restriction," in *Proc. IEEE Int. Conf. Auto. Sci. Eng.*, Washington DC, USA, 2008, pp. 103–108.

[37] X. W. Guo, S. X. Liu, M. C. Zhou, and G. D. Tian, "Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and Petri nets," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2435–2446, Nov. 2016.

[38] B. Huang, M. C. Zhou, P. Y. ZHang, and J. Yang, "Speedup techniques for multiobjective integer programs in designing optimal and structurally simple supervisors of AMS," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 48, no. 1, pp. 77–88, Jan. 2018.

[39] H.-Y. Jin and J. R. Morrison, "Transient scheduling of single armed cluster tools: Algorithms for wafer residency constraints," in *Proc. IEEE Int. Conf. Auto. Sci. Eng.*, Madison, WI, USA, 2013, pp. 856–861.

[40] J. G. Yi, S. W. Ding, D. Z. Song, and M. T. Zhang, "Steady-state throughput and scheduling analysis of multi-cluster tools for semiconductor manufacturing: a decomposition approach," *IEEE Trans. Automat. Sci. Eng.*, vol. 5, no. 2, pp. 321–336, 2008.

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

[42] J.-H. Lee, H.-J. Kim, and T.-E. Lee, "Scheduling lot switching operations for cluster tools," *IEEE Trans. Semiconduct. Manufact.*, vol. 26, no. 4, pp. 592–601, 2013.

[43] A. Davenport, "Integrated maintenance scheduling for semiconductor manufacturing," in *Proc. 7th Int. Conf. CPAIOR*, Bologna, Italy, 2010, pp. 92–96.

[44] J. A. Ramírez-Hernández, J. Crabtree, X. D. Yao, E. Fernandez, M. C. Fu, M. Janakiram, S. I. Marcus, M. O'Connor, and N. Patel, "Optimal preventive maintenance scheduling in semiconductor manufacturing systems: software tool and simulation case studies," *IEEE Trans. Semiconduct. Manufact.*, vol. 23, no. 3, pp. 477–489, Aug. 2010.

[45] D.-K. Kim, T.-E. Lee, and H.-J. Kim, "Optimal scheduling of transient cycles for single-armed cluster tools with parallel chambers," *IEEE Trans. Automat. Sci. Eng.*, vol. 13, no. 2, pp. 1165–1175, Apr. 2016.

[46] T.-K. Kim, C. Junga, and T.-E. Lee, "Scheduling start-up and close-down periods of dual-armed cluster tools with wafer delay regulation," *Int. J. Product. Res.*, vol. 50, no. 10, pp. 2785–2795, May 2012.

[47] Y. Qiao, N. Q. Wu, F. J. Yang, M. C. Zhou, and Q. H. Zhu, "Wafer sojourn time fluctuation analysis of time-constrained dual-arm cluster tools with wafer revisiting and activity time variation," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 48, no. 4, pp. 622–636, Apr. 2018.

[48] 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.

[49] 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, pp. 389–400, Mar. 2018.

[50] H.-J. Kim, J.-H. Lee, and T.-E. Lee, "Schedulability analysis for noncyclic operation of time-constrained cluster tools with time variation," *IEEE Trans. Automat. Sci. Eng.*, vol. 13, no. 3, pp. 1409–1414, Jul. 2016.

[51] M. Meyyappan, "A review of plasma enhanced chemical vapour deposition of carbon nanotubes," *J. Phys. D: Appl. Phys.*, vol. 42, no. 21, pp. 213001–213015, 2009.

[52] R. Muñoz and C. Gómez-Aleixandre, "Review of CVD synthesis of graphene," *Chem. Vap. Deposition*, vol. 19, no. 10-11-12, pp. 297–322, Dec. 2013.

[53] W. Liu, S. Kraemer, D. Sarkar, H. Li, P. M. Ajayan, and K. Banerjee, "Controllable and rapid synthesis of high-quality and large-area Bernal stacked bilayer graphene using chemical vapor deposition," *Chem. Mater.*, vol. 26, no. 2, pp. 907–915, Dec. 2013.

[54] I. Drobouchevitch, S. P. Sethi, and C. Sriskandarajah, "Scheduling dual gripper robotic cells: one-unit cycles," *Eur. J. Oper. Res.*, vol. 171, no. 2, pp. 598–631, Jun. 2006.

[55] X. W. Guo, S. X. Liu, M. C. Zhou, and G. D. Tian, "Dual-objective program and scatter search for the optimization of disassembly sequences subject to multi-resource constraints," *IEEE Trans. Automat. Sci. Eng.*, vol. 15, no. 3, pp. 1091–1013, Jul. 2018.

[56] X. H. Meng, J. Li, M. C. Zhou, X. Z. Dai, and J. P. Dou, "Population-based incremental learning algorithm for a serial colored traveling salesman problem," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 48, no. 2, pp. 277–288, Feb. 2018.

[57] J. Zhao, S. X. Liu, M. C. Zhou, X. W. Guo, and L. Qi, "Modified cuckoo search algorithm to solve economic power dispatch optimization problems,," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 4, pp. 794–806, May 2018.

[58] J. Zhao, S. X. Liu, M. C. Zhou, X. W. Guo, and L. Qi, "An improved binary cuckoo search algorithm for solving unit commitment problem: method description," *IEEE Access*, vol. 6, no. 4, pp. 43535–43545, May 2018.

**QingHua Zhu** (M'15–SM'17) received the Ph.D. degree in industrial engineering from Guangdong University of Technology, in 2013. During Oct. 2014 through Sep. 2015, and Sep. 2017 through Aug. 2018, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, USA. He joined Guangdong University of Technology, in 2003, and is currently an Associate Professor with the School of Computer Science and Technology. His research interests include scheduling and optimization, discrete event systems, and Petri nets.

**Yan Qiao** (M'16) received the B.S. and Ph.D. degrees in industrial engineering and mechanical engineering from Guangdong University of Technology, in 2009 and 2015, respectively. From Sep. 2014 to Sep. 2015, he was a Visiting Student with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, USA. From Jan. 2016 to Dec. 2017, he was a Post-Doctoral Research Associate with the Institute of Systems Engineering, Macau University of Science and Technology. Since Jan. 2018, he is an Assistant Professor at Macau University of Science and Technology. He has over 50 publications, including one book chapter and over 30 international journal papers. His research interests include discrete event systems, production planning, Petri nets, scheduling and control. Dr. Qiao was a recipient of the QSI Best Application Paper Award Finalist of 2011 IEEE International Conference on Automation Science and Engineering, the Best Student Paper Award of 2012 IEEE International Conference on Networking, Sensing and Control, and the Best Conference Paper Award Finalist of 2016 IEEE International Conference on Automation Science and Engineering. He has served as a reviewer for a number of journals.

**NaiQi Wu** (M'04–SM'05–F'19) received his B.S. degree in electrical engineering from Anhui University of Technology, in 1982, the M.S. and Ph.D. degrees in systems engineering both from Xi'an Jiaotong University, in 1985 and 1988, respectively. From 1988 to 1995, he was with Shenyang Institute of Automation, Chinese Academy of Sciences, and from 1995 to 1998, with Shantou University, He moved to Guangdong University of Technology in 1998. He joined Macau University of Science and Technology, in 2013. He was a visiting professor at Arizona State University, USA, in 1999; New Jersey Institute of Technology, USA, in 2004; University of Technology of Troyes, France, from 2007 to 2009; and Evry University, France, from 2010 to 2011. He is currently a Professor at the Institute of Systems Engineering, Macau University of Science and Technology. 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 160+ 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*, and is an Associate Editor of *Information Sciences*.

**Yan Hou** received the B.S. and M.S. degrees in computer science from Yangtze University, in 1999 and 2002, respectively, and the Ph.D. degree in industrial engineering from Guangdong University of Technology, in 2016. Since 2002, she has been with Guangdong University of Technology. Her research interests include production scheduling, information systems, and software engineering.