

Scheduling Dual-Arm Multi-Cluster Tools With Regulation of Post-Processing Time

Qinghua Zhu, *Senior Member, IEEE*, Bin Li, Yan Hou, Hongpeng Li, and Naiqi Wu, *Fellow, IEEE*

Abstract—As wafer circuit width shrinks down to less than ten nanometers in recent years, stringent quality control in the wafer manufacturing process is increasingly important. Thanks to the coupling of neighboring cluster tools and coordination of multiple robots in a multi-cluster tool, wafer production scheduling becomes rather complicated. After a wafer is processed, due to high-temperature chemical reactions in a chamber, the robot should be controlled to take it out of the processing chamber at the right time. In order to ensure the uniformity of integrated circuits on wafers, it is highly desirable to make the differences in wafer post-processing time among the individual tools in a multi-cluster tool as small as possible. To achieve this goal, for the first time, this work aims to find an optimal schedule for a dual-arm multi-cluster tool to regulate the wafer post-processing time. To do so, we propose polynomial-time algorithms to find an optimal schedule, which can achieve the highest throughput, and minimize the total post-processing time of the processing steps. We propose a linear program model and another algorithm to balance the differences in the post-processing time between any pair of adjacent cluster tools. Two industrial examples are given to illustrate the application and effectiveness of the proposed method.

Index Terms—Cluster tool, optimization, scheduling.

I. INTRODUCTION

SCHEDULING of multi-cluster tools has attracted much attention in recent years thanks to its popularity and complexity. A multi-cluster tool is a combination of several single-cluster tools. Typically, a single-cluster tool has a robot and several processing modules (PMs). The robot is surrounded by PMs to transport wafers. A dual-arm robot has two blades, which can hold two wafers at a time and import/export of wafers through the loadlock cassette mod-

ules (LLs). K ($K \geq 2$) single-cluster tools are integrated linearly into a K -cluster tool system. An example of a four-cluster tool is shown in Fig. 1. A buffering module (BM) links two adjacent cluster tools. It can temporarily accommodate incoming and outgoing wafers. The cluster tool with LLs is called the head tool and it is numbered the first one. In a steady state, the robot in the first cluster tool unloads a wafer from LLs, transfers it to the PMs for processing according to a predefined route as shown in Fig. 1, where it goes through the downstream tools and backs through the upstream tools one by one [1]. Finally, the robot in the head tool carries the completed wafer back to LLs. A swap strategy is adopted to schedule a robot for a dual-blade cluster tool. The dual blades with a swap strategy make the robot task time much shorter than the wafer processing time in a PM. Therefore, two-blade robots are more efficient than single-blade robots [2].

In semiconductor manufacturing [3], various constraints bring great challenges to scheduling cluster tools. During a chemical vapor deposition (CVD) process [4], a wafer must avoid excessive exposure to the mixed chemical gases at high temperatures in a PM. Thus, after a wafer is processed, it needs to limit the time during which it stays in a PM to avoid being scrapped, which is called wafer residency time constraints [5]–[10]. Furthermore, the circuit width on a wafer has been reduced to less than 10 nanometers under continuous development [11]. To fabricate high-end integrated-circuit chips, CVD equipment puts atom-thick layers of nanomaterials onto a wafer, which synthesizes monolayer, bilayer, or few-layer graphene. Consequently, as circuit width shrinks down, the quality of circuits on wafers is more susceptible to chemical reaction time in a chamber [12]. The wafer sojourn time in a chamber after its processing referred to as post-processing time affects the circuit quality. Wafer delay analysis has been done for single cluster tools [13]–[15]. It requires proper control on the post-processing time in different processing steps and the difference in post-processing time between two adjacent cluster tools as well.

Great efforts have made for modeling, analysis, and scheduling cluster tools [5], [13], [14] and [16]–[32] and deadlock analysis in flexible manufacturing systems [33]–[36]. The robotic manufacturing systems in [33]–[35] differ from the multi-cluster tools in the following aspects. First, the former does not cover residency time constraints and revisiting processes. Second, the former has one input station and another output station and it is configured in a linear layout; however, the latter has the same input and output station

Manuscript received April 25, 2022; revised June 24, 2022; accepted July 8, 2022. This work was supported in part by the National Natural Science Foundation of China (61673123), the Natural Science Foundation of Guangdong Province, China (2020A151501482), the Science and Technology development fund (FDCT), Macau SAR (0083/2021/A2, 0015/2020/AMJ), and Research Fund of Guangdong-Hong Kong-Macao Joint Laboratory for Intelligent Micro-Nano Optoelectronic Technology (2020B1212030010). Recommended by Associate Editor Shouguang Wang. (Corresponding author: Qinghua Zhu.)

Citation: Q. H. Zhu, B. Li, Y. Hou, H. P. Li, and N. Q. Wu, "Scheduling dual-arm multi-cluster tools with regulation of post-processing time," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 8, pp. 1730–1742, Aug. 2023.

Q. H. Zhu, B. Li, Y. Hou, and H. P. Li are with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China (e-mail: zhuqh@gdut.edu.cn; 2112005259@mail2.gdut.edu.cn; houyan@gdut.edu.cn; 2112005036@mail2.gdut.edu.cn).

N. Q. Wu is with the Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems, Macau University of Science and Technology, Macau 999078, China (e-mail: nqw@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.2023.123189

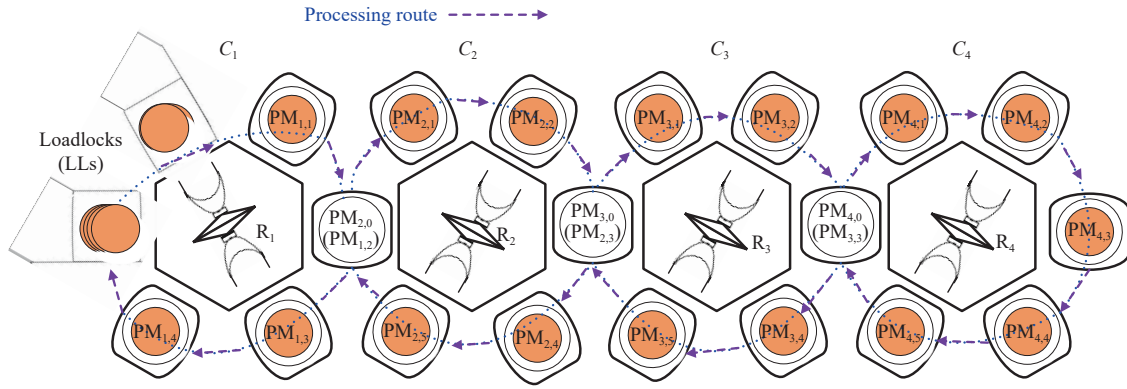


Fig. 1. A four-cluster tool. LLs are viewed as $PM_{1,0}$; $PM_{i,0}$ with $i > 1$ denotes a buffering module.

(loadlock modules) and it is configured in a circular layout. Due to these differences, the methods in [33]–[35] cannot solve the deadlock problems for multi-cluster tools.

A multi-cluster tool is said to be process-dominant if its bottleneck tool is process-bound. The study in [37] shows that a one-wafer cyclic schedule can be found for a process-dominant multi-cluster tool under a steady state [38]–[41], and [5] derives its schedulability conditions under wafer residency time constraints. Thus, one can effectively find an optimal cyclic schedule if a feasible one exists. Reference [42] proposes a non-cyclic scheduling method for multi-cluster tools. Now, it is rather significant to further examine the wafer post-processing time for multi-cluster tools. For the requirements of high-quality functional circuits, field-effect transistors must be built with high uniformity on a wafer from transistors to circuits [43], e.g., MoS_2 is grown by CVD, and the residual film is expected to deposit uniformly. Reducing the difference in post-processing time among a variety of process steps is conducive to keeping good uniformity among different layers of circuits on a wafer since a layer of circuits are typically fabricated by a processing step.

Differently from the aforementioned existing work, this study focuses on a K -cluster tool to get a better schedule with regulations on wafer post-processing time and high productivity. Reference [44] investigates the optimal scheduling of a dual-arm single-cluster tool with regulations of wafer post-processing time. However, its method and result cannot be applied to a multi-cluster tool.

For a K -cluster tool, it is very challenging to regulate the wafer post-processing time. The challenges come from the following facts. 1) Shortening wafer post-processing time at a given step requires that the robot changes its time to load/unload a wafer, resulting in the changes of time to perform loading/unloading actions at upstream and downstream steps including the buffer steps, which interact with the robot actions in adjacent cluster tools. Such interplay requires that delicate control of the cycle time should be made to coordinate multiple robots' actions to keep the cycle time of a K -cluster tool constant. If such coordination is not well conducted, there would be cycle time fluctuation that easily prop-

agates to its adjacent cluster tools and affects their cycle time. 2) Narrowing the difference in post-processing time between adjacent tools is a complex issue because it is very delicate to coordinate multiple robots to pace each tool and each step in a single tool.

The main contributions of this paper are twofold as follows.

1) In order to realize the adjustment of post-processing time, we propose effective algorithms to minimize the total post-processing time and balance its difference among the processing steps inside each tool.

2) We propose a linear programming model and a condition table to find the upper bound for adjusting the post-processing time in adjacent cluster tools. Then, an efficient algorithm is designed to adjust each cluster tool to balance the difference in post-processing time between adjacent cluster tools.

The rest of this article is structured as follows. Section II analyzes the temporal properties of scheduling multi-cluster tools. Section III proposes algorithms to find an optimal schedule that can minimize post-processing time and avoid uneven post-processing time among the processing steps or two adjacent cluster tools. Section IV gives two examples for applying the proposed method. Section V makes a summary of this work.

II. PROBLEM DESCRIPTION

A. Operation and Properties of a K -Cluster Tool

Following [27] and [38], we make the following assumptions for a dual-arm K -cluster tool ($K \geq 2$).

1) Two LLs can continuously provide enough wafers for PMs without interruption. Thus, a K -cluster tool can run under a steady state.

2) A BM can hold one wafer at a time without a processing function.

3) Every processing step is configured with a PM. Thereafter, in order to facilitate the presentation, we refer to a PS as a PM. It is obvious that a wafer is processed at a PM once and enters a BM twice.

Let $N_q = \{0, 1, 2, 3, \dots, q\}$ and $N_q^+ = N_q \setminus \{0\}$, where q is a given positive integer. Then, C_i denotes the i th cluster tool and R_i denotes the robot in C_i , $i \in N_K^+$. The buffering step in C_i at a buffering module can be represented by $b[i]$, $i \in N_{K-1}^+$. Let $b[K] = \emptyset$. Let $n[i]$ denote the index of the last step in C_i . Then,

the processing steps of C_i are denoted by $PS_{i,0}, PS_{i,1}, \dots, PS_{i,b[i]}, \dots$, and $PS_{i,n[i]}$. As shown in Fig. 1, we can get the wafer processing route in a K -cluster tool as $\langle LL \rightarrow PM_{1,1} \rightarrow \dots \rightarrow PM_{1,b[1]} (PM_{2,0}) \rightarrow PM_{2,1} \rightarrow \dots \rightarrow PM_{2,b[1]} (PM_{3,0}) \rightarrow \dots \rightarrow PM_{K-1,b[K-1]} (PM_{K,0}) \rightarrow PM_{K,1} \rightarrow \dots \rightarrow PM_{K,n[K]} \rightarrow \dots \rightarrow PM_{K,0} (PM_{K-1,b[K-1]}) \rightarrow PM_{K-1,b[K-1]+1} \rightarrow \dots \rightarrow PM_{3,0} (PM_{2,b[2]}) \rightarrow PM_{2,b[2]+1} \rightarrow \dots \rightarrow PM_{2,0} (PM_{1,b[1]}) \rightarrow \dots \rightarrow PM_{1,n[1]} \rightarrow LL \rangle$. For C_i , let $\omega_{i,j}$ and $\omega_{i,jS}$ denote the robot waiting time before unloading and loading a wafer from/into $PM_{i,j}$, respectively. For R_i , it takes μ_i time units to move between two PMs or a PM and an LL, λ_i time units to load/unload a wafer into/from a PM (LL), and ρ_i time units to make a rotation. Let $\alpha_{i,j}$ denote the wafer processing time at $PM_{i,j}$. The processing time at buffering modules and LLs is zero due to no functional processing there. The symbols for timeliness analysis are listed in Table I. In addition, the time required in a K -cluster tool is shown in Table II.

TABLE I
SYMBOLS FOR TIMELINESS ANALYSIS

Symbol	Meaning
N_q	$= \{0, 1, 2, 3, \dots, q\}$, where q is a positive integer.
N_q^+	$= \{1, 2, 3, \dots, q\}$.
C_i	The i th cluster tool.
R_i	The robot in C_i .
$PM_{i,j}$	The j th process module in C_i .
$n[i]$	The index of the last step in C_i .
$b[i]$	Index of the buffering step in C_i .
λ_i	Time taken by R_i for unloading/loading a wafer.
μ_i	Time taken by R_i for moving from one PM to another.
ρ_i	Time taken by R_i 's rotation.
α_{ij}	The time of processing a wafer at $PM_{i,j}$.
$\theta_{i,j}$	Cycle time of $PM_{i,j}$.
$\delta_{i,j}$	The permissive longest time for a wafer to stay at $PM_{i,j}$ after its processing.
$\xi_{i,j}$	The time needed for completing a wafer at $PM_{i,j}$.
$\Pi_{i,jL}$	The lower bound of $\theta_{i,j}$.
$\Pi_{i,jU}$	The upper bound of $\theta_{i,j}$.
Θ_i	The cycle time of C_i .
Θ	The cycle time of a K -cluster tool.
ψ_i	Robot R_i 's cycle time.

TABLE II
DECISION VARIABLES AND THEIR ASSOCIATED VARIABLES

Symbol	Meaning
$\omega_{i,j}$	R_i 's waiting time before unloading a wafer from $PM_{i,j}$.
$\omega_{i,jS}$	R_i 's waiting time before loading a wafer into $PM_{i,j}$.
$\tau_{i,j}$	Wafer sojourn time at $PM_{i,j}$.
$r_{i,j}$	The post-processing time of a wafer at $PM_{i,j}$.

B. Temporal Properties of Individual Tools

By a swap strategy, robot R_i executes the following sequence of operations in C_i

$\langle \text{Moving to } PM_{i,0} \text{ (it takes } \mu_i \text{ time units; hereafter, only the amount of time is presented)} \rightarrow \text{waiting there for } \omega_{i,0} \text{ time units} \rightarrow \text{unloading a wafer from } PM_{i,0} (\lambda_i) \rightarrow \text{rotating } (\rho_i) \rightarrow \text{waiting there for } \omega_{i,0S} \text{ time units} \rightarrow \text{loading a wafer gripped by the other blade into } PM_{i,0} (\lambda_i) \rightarrow \dots \rightarrow \text{moving to } PM_{i,b[i]} (\mu_i) \rightarrow \text{waiting there for } \omega_{i,b[i]} \text{ time units} \rightarrow \text{unloading a wafer from } PM_{i,b[i]} (\lambda_i) \rightarrow \text{rotating } (\rho_i) \rightarrow \text{waiting there for } \omega_{i,b[i]S} \text{ time units} \rightarrow \text{loading a wafer gripped by the other blade into } PM_{i,b[i]} (\lambda_i) \rightarrow \dots \rightarrow \text{moving to } PM_{i,n[i]} (\mu_i) \rightarrow \text{waiting there for } \omega_{i,n[i]} \text{ time units} \rightarrow \text{unloading the finished wafer from } PM_{i,n[i]} (\lambda_i) \rightarrow \text{rotating } (\rho_i) \rightarrow \text{waiting there for } \omega_{i,n[i]S} \text{ time units} \rightarrow \text{loading a wafer gripped by the other blade into } PM_{i,n[i]} (\lambda_i) \rightarrow \text{moving to } PM_{i,0} (\mu_i) \text{ again} \rangle$.

Hence, robot R_i 's ($2 \leq i \leq K$) cycle time is

$$\begin{aligned} \psi_i &= (n[i] + 1)(2\lambda_i + \mu_i + \rho_i) + \sum_{j=0}^{n[i]} \omega_{i,jS} + \sum_{j=0}^{n[i]} \omega_{i,j} \\ &= \psi_{i,1} + \psi_{i,2} \end{aligned} \quad (1)$$

where $\psi_{i,1} = (n[i] + 1)(2\lambda_i + \mu_i + \rho_i)$, $i \in \mathbf{N}_K^+ \setminus \{1\}$.

Since LLs in C_1 have no capacity limit, robot R_1 can directly load/unload wafers with the same blade. Hence, robot R_1 's cycle time is

$$\begin{aligned} \psi_1 &= (n[1] + 1)(2\lambda_1 + \mu_1) + n[1] \times \rho_1 + \sum_{j=0}^{n[1]} \omega_{1,jS} + \sum_{j=0}^{n[1]} \omega_{1,j} \\ &= \psi_{1,1} + \psi_{1,2} \end{aligned} \quad (2)$$

where $\psi_{1,1} = (n[1] + 1)(2\lambda_1 + \mu_1) + n[1] \times \rho_1$.

Therefore, by (1) and (2), a robot's cycle time consists of its task time and waiting time. The former is a constant that is equal to $\psi_{i,1}$, while the latter is $\psi_{i,2} = \sum_{j=0}^{n[i]} \omega_{i,j} + \sum_{j=0}^{n[i]} \omega_{i,jS}$.

The key to scheduling a multi-cluster tool is to coordinate the operations of the multiple robots. To do so, we need to analyze its temporal properties. The time is taken to complete a wafer at $PS_{i,j}$, $i \in \mathbf{N}_K^+$, as follows:

$$\xi_{i,j} = \alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}, \quad j \in \mathbf{N}_{n[i]}^+ \quad (3)$$

and

$$\xi_{i,0} = 2\lambda_i + \rho_i + \omega_{i,0S}. \quad (4)$$

In (3) and (4), $\omega_{i,jS}$ is a decision variable, while λ_i, ρ_i , and $\alpha_{i,j}$ are given parameters. When $\omega_{i,jS}$ is zero, we get the shortest time for completing a wafer at $PM_{i,j}$, i.e., the lower bound $\Pi_{i,jL}$, $i \in \mathbf{N}_K^+$, as

$$\Pi_{i,jL} = \alpha_{i,j} + 2\lambda_i + \rho_i, \quad j \in \mathbf{N}_{n[i]}^+ \quad (5)$$

and

$$\Pi_{i,0L} = 2\lambda_i + \rho_i. \quad (6)$$

After a wafer is processed at $PM_{i,j}$, it cannot stay at $PM_{i,j}$ for more than $\delta_{i,j} \geq 0$ time units. Thus, the upper bound of $\Pi_{i,j}$ is

$$\Pi_{i,jU} = \alpha_{i,j} + \delta_{i,j} + 2\lambda_i + \rho_i, \quad j \in \mathbf{N}_{n[i]}. \quad (7)$$

The time between loading a wafer into $\text{PM}_{i,j}$ and unloading it from there is called wafer sojourn time, which is denoted by $\tau_{i,j}$. Obviously, $\alpha_{i,j} + \delta_{i,j} \geq \tau_{i,j} \geq \alpha_{i,j}$ should be met, which means that a wafer staying at a PM subject to such a constraint leads to a feasible schedule. The cycle time at $\text{PS}_{i,j}$ is

$$\theta_{i,j} = \tau_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}, \quad j \in \mathbf{N}_{n[i]}^+. \quad (8)$$

Although BMs and LLs perform no functional processing, i.e., $\alpha_{i,0} = 0$, a wafer may stay at BMs or LLs for $\tau_{i,0}$ time units. We get that the cycle time at $\text{PS}_{i,0}$ is

$$\theta_{i,0} = \tau_{i,0} + 2\lambda_i + \rho_i + \omega_{i,0S}. \quad (9)$$

$\tau_{i,j}$ can be calculated as follows:

$$\tau_{i,j} = \psi_i - (2\lambda_i + \rho_i + \omega_{i,jS}), \quad j \in \mathbf{N}_{n[i]}. \quad (10)$$

The above equation shows that the wafer residency time depends on robot waiting time $\omega_{i,jS}$ when ψ_i is given.

Let $\Pi_i = \max\{\Pi_{i,0L}, \Pi_{i,1L}, \dots, \Pi_{i,n[i]L}, \psi_{i,1}\}$ and Θ_i denote the cycle time of C_i . Then, if $\Pi_i \neq \psi_{i,1}$, C_i is process-bound. To get a cyclic schedule, C_i should be scheduled to satisfy

$$\Theta_i = \psi_i = \theta_{i,j} = \xi_{i,j}, \quad j \in \mathbf{N}_{n[i]} \setminus \{0, b[i]\}. \quad (11)$$

Equations (8) and (9) show that $\omega_{i,jS}$ is a decision variable for cycle time $\theta_{i,j}$ at $\text{PS}_{i,j}$ with $j \in \mathbf{N}_{n[i]}$. Notice that (1), (2) and (8) contain decision variables $\omega_{i,jS}$. Thus, (11) can be made to be satisfied by adjusting R_i 's waiting time $\omega_{i,jS}$. Suppose that Θ is the given cycle time for a K -cluster tool and let $\Theta_i = \psi_i = \theta_{i,j} = \Theta$. After $\omega_{i,jS}$ is determined to meet (8) and (9), the robot's remaining idle time ($\psi_{i,2} - \sum_{j=0}^{n[i]} \omega_{i,jS}$) can be assigned to $\omega_{i,j}$'s.

Let $r_{i,j}$ denote the wafer post-processing time at $\text{PM}_{i,j}$, which is the time between the time when its processing ends and the time when R_i starts to unload the wafer. Thus, the post-processing time at $\text{PM}_{i,j}$, $i \in \mathbf{N}_K^+$ is

$$r_{i,j} = \tau_{i,j} - \alpha_{i,j}, \quad j \in \mathbf{N}_{n[i]} \setminus \{0, b[i]\}. \quad (12)$$

If the robot waiting time $\omega_{i,jS}$ is properly set, (11) can be satisfied. It implies that the cycle time of each cluster tool, each robot, and each PM is equal so that a multi-cluster tool can yield one wafer within a cycle. Given cycle time $\Theta_i = \theta_{i,j} = \Theta$ with $i \in \mathbf{N}_K^+$ and $j \in \mathbf{N}_{n[i]}$, by (8) and (9), if $\omega_{i,jS}$ ($i \in \mathbf{N}_K^+$) increases, then wafer sojourn time $\tau_{i,j}$ decreases, and by (12), $r_{i,j}$ decreases as well. In the view of tool C_i , minimizing the total post-processing time $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ is conducive to producing high-quality circuit chips on wafers.

C. Problem Definition

The objective of this work is to regulate wafer post-processing time such that high-quality circuits can be ensured on a wafer. Therefore, this work aims to accomplish the following missions.

- 1) Minimize the cycle time of a dual-arm K -cluster tool.
- 2) Minimize $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ for C_i , $i \in \mathbf{N}_K^+$ and shorten the difference in post-processing time among $\text{PS}_{i,j}$, $j \in \mathbf{N}_{n[i]} \setminus \{0, b[i]\}$.
- 3) Shorten the difference in post-processing time between adjacent cluster tools.

III. SCHEDULING ALGORITHMS

A. Coordination of Multiple Robots

The minimal cycle time of a process-bound dual-arm cluster tool C_i can be calculated as $\Pi_i = \max\{\Pi_{i,0L}, \Pi_{i,1L}, \dots, \Pi_{i,n[i]L}\}$, $i \in \mathbf{N}_K^+$. When a K -cluster tool runs, multiple robots can be coordinated such that the cycle time of R_i is the same as C_i , i.e., $\psi_i = \Pi_i$.

Let $\Pi = \max\{\Pi_1, \Pi_2, \dots, \Pi_K\}$. As aforementioned, Θ is the cycle time of a K -cluster tool. Assume that C_g ($2 \leq g \leq K-1$) has the heaviest workload among the K individual tools. Thus, we have $\Pi = \Pi_g$, and the bottleneck of a K -cluster tool is C_g . For a process-dominant K -cluster tool, to obtain a one-wafer cyclic schedule, we must have

$$\Theta_1 = \Theta_2 = \dots = \Theta_K = \Theta. \quad (13)$$

This means that each cluster tool should be scheduled with the same cycle time. The cycle time must satisfy $\Theta \geq \Pi = \Pi_g$. Under this premise, it can guarantee to apply a swap strategy to each C_i , $i \in \mathbf{N}_K^+$.

Given the cycle time, the wafer processing time and the robot activity time are deterministic, the post-processing time and robot waiting time are variables, and interact with each other. Therefore, it is necessary to ensure that the robot waiting time is coordinated so that an individual tool can be scheduled at the same pace without conflict over its buffering module, and the difference in post-processing time among different steps can be regulated as small as possible. Note that a BM does not participate in the processing of wafers. Thus, BMs have no wafer residency time constraints.

Algorithm 1 Determine the Robot Waiting Time for a K -Cluster Tool

Input: $\lambda_i, \mu_i, \alpha_{i,j}, \delta_{i,j}, \rho_i$

Output: $\Theta, \omega_{i,jS}, \omega_{i,j}$

- 1: Calculate $\psi_{i,1}, \psi_{i,2}, \Pi_{i,jL}, \Pi_{i,jU}$, by (1), (2) and (5)–(7), respectively.
 - 2: $\Pi_i \leftarrow \max\{\Pi_{i,0L}, \Pi_{i,1L}, \dots, \Pi_{i,n[i]L}\}$ for $i \in \mathbf{N}_K^+$.
 - 3: $\Theta \leftarrow \max\{\Pi_1, \Pi_2, \dots, \Pi_K\}$
 - 4: **For** $i=1$ to K **do**
 - 5: **If** $\psi_{i,1} \leq \Theta$ and $\Theta \leq \Pi_{i,jU}$, $j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}$ **Then**
 - 6: Call Algorithm 2 /*Case 2
 - 7: **Else if** $\Pi_{i,jL} \leq \psi_{i,1} \leq \Pi_{i,jU}$, $j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}$ **Then** /*Case 1
 - 8: $\omega_{i,jS} \leftarrow 0, \omega_{i,j} \leftarrow 0$, for $j \in \mathbf{N}_{n[i]}$
 - 9: **Else if** $\cap_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} [\Pi_{i,jL}, \Pi_{i,jU}] = \emptyset$ **Then**
 - 10: Call Algorithm 3 /*Case 3
 - 11: **Else**
 - 12: **Return** /*Unschedulable
 - 13: **Endif**
 - 14: **Endif**
 - 15: **Endif**
 - 16: **If** $i \neq K$ and $\tau_{i,b[i]} < 2\lambda_{i+1} + \rho_{i+1}$, **Then**
 - 17: **Return** /*Unschedulable
 - 18: **End if**
 - 19: **EndFor**
 - 20: **End**
-

Algorithm 2 Set R_i 's Waiting Time for Case 2

Input: $\lambda_i, \mu_i, \alpha_{i,j}, \delta_{i,j}, \Theta, \rho_i$
Output: $\omega_{i,j}, \omega_{i,jS}, r_{i,j}$

```

1:  $V \leftarrow \{j \mid j \neq 0 \text{ and } j \neq b[i], j \in \mathbf{N}_{n[i]}\}$ 
2:  $\varepsilon \leftarrow \sum_{j \in V} (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i))$ 
3: If  $\psi_{i,2} \geq \varepsilon$ 
4:    $h \leftarrow \psi_{i,2} - \varepsilon$ 
5:   For  $j \in V$  do
6:      $r_{i,j} \leftarrow 0, \omega_{i,jS} \leftarrow \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i), \omega_{i,j} \leftarrow h/|V|$ 
7:   Endfor
8: Else  $\psi_{i,2} < \varepsilon$ 
9:    $h \leftarrow \varepsilon - \psi_{i,2}$ 
10:  While  $V \neq \emptyset$  Then
11:     $\Delta \leftarrow h/|V|, A \leftarrow \emptyset, B \leftarrow \emptyset$ 
12:    For each  $j \in V$  do
13:      If  $\Delta \leq \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)$  Then  $A \leftarrow A \cup \{j\}$ 
14:      Else  $B \leftarrow B \cup \{j\}$ 
15:    Endif
16:  Endfor
17:  If  $B \neq \emptyset$  Then
18:    For each  $j \in B$  do
19:       $r_{i,j} \leftarrow \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i), \omega_{i,jS} \leftarrow 0, \omega_{i,j} \leftarrow 0$ 
20:       $h \leftarrow h - (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}))$ 
21:    EndFor
22:     $V \leftarrow V \setminus B$ 
23:  Else
24:    For each  $j \in A$  do
25:       $r_{i,j} \leftarrow \Delta$ 
26:       $\omega_{i,jS} \leftarrow \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) - r_{i,j}, \omega_{i,j} \leftarrow 0$ 
27:    EndFor
28:     $V \leftarrow \emptyset$ 
29:  EndIf
30: Endwhile
31: EndIf
32:  $\omega_{i,0S} \leftarrow 0, \omega_{i,0} \leftarrow 0, \omega_{i,b[i]S} \leftarrow 0, \omega_{i,b[i]} \leftarrow 0$ 
33: End

```

It follows from [24] that if $\Theta \geq \Pi$ is the cycle time of a process-dominant K -cluster tool with wafer residency time constraints, a one-wafer periodic schedule exists if and only if, for any pair of C_i and C_{i+1} , $i \in \mathbf{N}_{K-1}^+$, $\omega_{i,jS}$'s and $\omega_{(i+1),jS}$'s are set such that the following conditions are satisfied:

$$\theta_{i,j} = \theta_{(i+1),f} = \Theta, \quad j \in \mathbf{N}_{n[i]} \text{ and } f \in \mathbf{N}_{n[i+1]} \quad (14)$$

$$\tau_{i,j} \in [\alpha_{i,j}, \alpha_{i,j} + \delta_{i,j}], \quad i \in \mathbf{N}_K^+ \text{ and } j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\} \quad (15)$$

$$\tau_{i,b[i]} \geq 2\lambda_{i+1} + \rho_{i+1} + \omega_{(i+1),0S} \quad (16)$$

and

$$\tau_{(i+1),0} \geq 2\lambda_i + \rho_i + \omega_{i,b[i]S}. \quad (17)$$

B. Algorithms

By (10), (12) and (15)–(17), setting appropriate $\omega_{i,jS}$'s can render conditions (15)–(17) satisfied and further shorten wafer post-processing time. If $\omega_{i,jS}$'s are determined, a one-wafer cyclic schedule is found. Therefore, we propose Algorithms 1–3 to calculate $\omega_{i,jS}$'s to obtain a schedule, where the cardi-

nality of set V is denoted by $|V|$. We propose Algorithm 1 to handle three different workload cases for each cluster tool. Algorithm 1 determines the condition to be satisfied by a K -cluster tool. Section IV details the calculation process for Algorithm 1.

Case 1: $\Pi_{i,jL} \leq \psi_{i,1} \leq \Pi_{i,jU}$, $i \in \mathbf{N}_K^+$, indicates that robot R_i is always busy and tool C_i is transport-bound. Therefore, the robot waiting time is zero as shown in Line 8 of Algorithm 1.

Case 2: $\psi_{i,1} \leq \Theta$ and $\Theta \leq \Pi_{i,jU}$, $i \in \mathbf{N}_K^+$, means that the workloads of all processing steps in C_i are relatively balanced and tool C_i is process-bound.

The following results are given to show the optimality of the schedule obtained by Algorithm 2.

Theorem 1: Given $\Theta = \Pi_g$ as the cycle time for a dual-arm cluster tool, if and only if $\psi_{i,1} \leq \Theta$ and $\Theta \leq \Pi_{i,jU}$, $i \in \mathbf{N}_K^+$, are satisfied, Algorithm 2 finds a schedule that achieves the lower bound of its cycle time and minimizes the total post-processing time.

Proof: Let $\Theta = \Pi_g$ be the cycle time of a K -cluster tool. We examine whether a feasible schedule with cycle time Θ can be found and whether the post-processing time is minimal.

By Lines 1 and 2, we have $\varepsilon = \sum_{j \in V} (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i))$. If $\psi_{i,2} \geq \varepsilon$, we set $\omega_{i,jS} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)$, $j \in V$. Then, we have $\tau_{i,j} = \Theta - (2\lambda_i + \rho_i + \omega_{i,jS})$, $j \in V$. The post-processing time at step j in C_i is $r_{i,j} = \tau_{i,j} - \alpha_{i,j} = 0 < \delta_{i,j}$, $j \in V$. Thus, $r_{i,j}$ is minimized, and the wafer residency time constraints are satisfied. If $\psi_{i,2} < \varepsilon$, we have $h = \psi_{i,2} - \varepsilon$. By Line 11, $\Delta = h/|V|$. Lines 13 and 14 show that if $\Delta \leq \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)$ holds, Step j is put into set A , otherwise, Step j is put into set B .

If $B \neq \emptyset$, $r_{i,j} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)$ and $\omega_{i,jS} = 0$ for $j \in B$, Lines 18–22 deal with this situation. We have $r_{i,j} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) \geq \Pi_{i,jL} - (\alpha_{i,j} + 2\lambda_i + \rho_i) = 0$ and $r_{i,j} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) \leq \Pi_{i,jU} - (\alpha_{i,j} + 2\lambda_i + \rho_i) = \delta_{i,j}$. Then, $h = \Theta - r_{i,j}$. Because of $\Delta = h/|V|$ and $\Delta > \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)$, we have $h = \Delta|V| > \sum_{j \in B} (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i))$. Therefore, $h > 0$ always holds. By Line 22, $V = V \setminus B$.

If $B = \emptyset$, $r_{i,j} = \Delta$ and $\omega_{i,jS} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) - r_{i,j}$ for $j \in A$, Lines 24–28 deal with this situation. We have $r_{i,j} = \Delta = h/|V| \leq \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) \leq \Pi_{i,jU} - (\alpha_{i,j} + 2\lambda_i + \rho_i) = \delta_{i,j}$ and $r_{i,j} > 0$ because of $\psi_{i,2} < \varepsilon$. Since $\sum_{j \in V} \omega_{i,jS} = \sum_{j \in V} (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)) - \sum_{j \in V} r_{i,j} = \psi_{i,2}$, we ensure that the value of $\psi_{i,2}$ has been assigned to $\omega_{i,jS}$'s ($j \in V$), i.e., $\sum_{j \in V} \omega_{i,jS}$ is maximized and $\sum_{j \in V} r_{i,j}$ is minimized. ■

Algorithm 2 deals with the processing steps in $V = \{j \mid j \neq 0 \text{ and } j \neq b[i], j \in \mathbf{N}_{n[i]}\}$. In the case of $\varepsilon \geq \psi_{i,2}$, we have that, for any step $j \in V$, the available idle time for robot R_i 's waiting is enough so that the post-processing time is zero and it is unnecessary to handle this situation. In the case of $\psi_{i,2} < \varepsilon$, it is easy to show that, for step $j \in V$, R_i 's available idle time for waiting is not enough so that the post-processing time should be evenly distributed. Then, Algorithm 2 solves the above two cases. Lines 1–8 handle the case of $\varepsilon \geq \psi_{i,2}$. Lines 10–33 deal with the case of $\varepsilon < \psi_{i,2}$ to adjust the post-processing time evenly. In addition, Lines 4 and 9 can ensure the most effective distribution of waiting time and minimize the total post-processing time.

Algorithm 3 Set R_i 's Waiting Time for Case 3

Input: $\lambda_i, \mu_i, \alpha_{i,j}, \delta_{i,j}, \Theta, \rho_i$
Output: $\omega_{i,j}, \omega_{i,jS}, r_{i,j}$

```

1:  $E \leftarrow \{j \mid \Pi_{i,jU} < \Theta, j \neq 0, j \neq b[i], j \in \mathbf{N}_{n[i]}\}$ 
2:  $F \leftarrow \mathbf{N}_{n[i]} \setminus (E \cup \{0, b[i]\}), V = E \cup F$ 
3:  $\omega_{i,jS} \leftarrow \Theta - (\alpha_{i,j} + \delta_{i,j} + 2\lambda_i + \rho_i), \text{ for } j \in E$ 
4:  $\omega_{i,jS} \leftarrow 0, \text{ for } j \in F$ 
5: If  $\sum_{j \in E} \omega_{i,jS} > \psi_{i,2}$  Then
6:   Return /* Unschedulable
7: EndIf
8:  $\varepsilon \leftarrow \psi_{i,2} - \sum_{j \in E} \omega_{i,jS}$ 
9:  $\eta \leftarrow \sum_{j \in F} (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)) + \sum_{j \in E} \delta_{i,j}$ 
10: If  $\varepsilon \geq \eta$ 
11:  $h \leftarrow \eta$ 
12:   For each  $j \in V$  do
13:      $r_{i,j} \leftarrow 0, \omega_{i,jS} \leftarrow \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i), \omega_{i,j} \leftarrow h/|V|$ 
14:   EndFor
15: Else /*  $\varepsilon < \eta$ 
16:    $h \leftarrow \eta - \varepsilon$ 
17:   While  $V \neq \emptyset$  Then
18:      $\Delta \leftarrow h/|V|, A \leftarrow \emptyset, B \leftarrow \emptyset$ 
19:     For each  $j \in V$  do
20:       If  $\Delta \leq \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS})$  Then  $A \leftarrow A \cup \{j\}$ 
21:       Else  $B \leftarrow B \cup \{j\}$ 
22:     EndIf
23:   Endfor
24:   If  $B \neq \emptyset$  Then
25:     For each  $j \in B$  do
26:        $r_{i,j} \leftarrow \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}),$ 
27:        $\omega_{i,jS} \leftarrow 0, \omega_{i,j} \leftarrow 0$ 
28:        $h \leftarrow h - (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}))$ 
29:     Endfor
29:      $V \leftarrow V \setminus B$ 
30:   Else
31:     For each  $j \in A$  do
32:        $r_{i,j} \leftarrow \Delta$ 
33:        $\omega_{i,jS} \leftarrow \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) - r_{i,j}, \omega_{i,j} \leftarrow 0$ 
34:     EndFor
35:      $V \leftarrow \emptyset$ 
36:   EndIf
37: Endwhile
38: EndIf
39:  $\omega_{i,0S} \leftarrow 0, \omega_{i,0} \leftarrow 0, \omega_{i,b[i]S} \leftarrow 0, \omega_{i,b[i]} \leftarrow 0$ 
40: End

```

In Algorithm 2, all the statements make calculations based on closed-form expressions. The number of iterations in the For-loop of Lines 5–7 is no more than $n[i]$. The number of iterations in the For-loop of Lines 10–30 is no more than $(n[i])^2$. It is obvious that the computational complexity of Algorithm 2 is polynomial.

Case 3: $\cap_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} [\Pi_{i,jL}, \Pi_{i,jU}] = \emptyset, i \in \mathbf{N}_K^+$, means that the workloads of the processing steps are unbalanced. In this case, Algorithm 3 is proposed to find a feasible schedule so that each processing step can achieve the cycle time of a K -cluster tool.

Theorem 2: If $\cap_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} [\Pi_{i,jL}, \Pi_{i,jU}] = \emptyset$ holds with $i \in \mathbf{N}_K^+$, Algorithm 3 can determine whether a feasible schedule exists. If so, Algorithm 3 finds a schedule that minimizes the total post-processing time.

Proof: By Lines 1 and 2, we have $E = \{j \mid \Pi_{i,jU} < \Theta, j \neq 0, j \neq b[i], j \in \mathbf{N}_{n[i]}\}, F = \mathbf{N}_{n[i]} \setminus E \cup \{0, b[i]\}$, and $V = E \cup F$. For $j \in E$, we set $\omega_{i,jS} = \Theta - (\alpha_{i,j} + \delta_{i,j} + 2\lambda_i + \rho_i)$. Hence, we have $\tau_{i,j} = \Theta - (2\lambda_i + \rho_i + \omega_{i,jS}) = \alpha_{i,j} + \delta_{i,j}$. Then, by Line 5, if $\sum_{j \in E} \omega_{i,jS} \leq \psi_{i,2}$, all steps in E can satisfy the wafer residency time constraints. If $\sum_{j \in E} \omega_{i,jS} > \psi_{i,2}$, $\exists i \in E$ such that $\omega_{i,jS} < \Theta - (\alpha_{i,j} + \delta_{i,j} + 2\lambda_i + \rho_i)$ holds, which leads to $\tau_{i,j} > \alpha_{i,j} + \delta_{i,j}$. Therefore, tool C_i is not schedulable, which is given by Line 6 in Algorithm 3.

By Lines 8 and 9, we have $\varepsilon = \psi_{i,2} - \sum_{j \in E} \omega_{i,jS}$ and $\eta = \sum_{j \in F} (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)) + \sum_{j \in E} \delta_{i,j}$.

If $\varepsilon \geq \eta$, then Lines 10–14 present that $\omega_{i,jS} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i), j \in V$. We have $\tau_{i,j} = \Theta - (2\lambda_i + \rho_i + \omega_{i,jS}) = \alpha_{i,j}, j \in V$. The post-processing time at $\text{PM}_{i,j}$ is $r_{i,j} = \tau_{i,j} - \alpha_{i,j} = 0 < \delta_{i,j}, j \in V$. Thus, $r_{i,j}$ is minimized, and the wafer residency time constraints are satisfied.

If $\varepsilon < \eta$, we have $h = \varepsilon - \eta$. By Line 18, $\Delta = h/|V|$. Lines 19–22 show that for $j \in V$, if $\Delta \leq \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS})$ holds, Step j is put into set A , otherwise, it is put into set B . Lines 24–29 set the post-processing time and robot waiting time at steps in B .

If $B \neq \emptyset, r_{i,j} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS})$ and $\omega_{i,jS} = 0$ for $j \in B$. We have $r_{i,j} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}) \geq \Pi_{i,jL} - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}) = 0$ and $r_{i,j} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}) \leq \Pi_{i,jU} - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}) = \delta_{i,j}$. Then, $h = h - r_{i,j}$. Because of $\Delta = h/|V|$ and $\Delta > \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}), h = \Delta|V| > \sum_{j \in B} (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}))$ holds. Therefore, $h > 0$ always holds. By Line 29, $V = V \setminus B$.

If $B = \emptyset$, Lines 31–35 set robot waiting time and post-processing time at steps in A . We have $r_{i,j} = \Delta$ and $\omega_{i,jS} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) - r_{i,j}$ for $j \in A$. Then, $r_{i,j} = \Delta = h/|V| \leq \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}) \leq \Pi_{i,jU} - (\alpha_{i,j} + 2\lambda_i + \rho_i + \omega_{i,jS}) = \delta_{i,j}$ and $r_{i,j} > 0$ because of $\psi_{i,2} < \varepsilon$. Since $\sum_{j \in V} \omega_{i,jS} = \sum_{j \in V} (\Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i)) - \sum_{j \in V} r_{i,j} = \psi_{i,2}$, we have that the value of $\psi_{i,2}$ has been assigned to $\omega_{i,jS}$'s ($j \in V$), i.e., $\sum_{j \in V} \omega_{i,jS}$ is maximized and $\sum_{j \in V} r_{i,j}$ is minimized. ■

Algorithm 3 deals with the situation $\cap_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} [\Pi_{i,jL}, \Pi_{i,jU}] = \emptyset, i \in \mathbf{N}_K^+$, which implies that some steps have a heavier workload than other steps. Given the cycle time of Θ , the robot waiting time is pre-allocated in E and F . Then, check whether a cluster tool is schedulable or not in Line 5. After that, according to the workloads of a robot and processing steps, Lines 10–40 can assign post-processing time evenly which is also minimized. Notice that Condition $\cap_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} [\Pi_{i,jL}, \Pi_{i,jU}] = \emptyset$ for Algorithm 3 indicates that the workloads of the processing steps are significantly unbalanced. That is to say, the processing steps in E are much faster than others and a feasible schedule can be obtained only by carefully setting the robot waiting time at different steps.

In Algorithm 3, all the statements make the calculations based on closed-form expressions. The number of iterations in the For-loop of Lines 3 and 4 is no more than $n[i]$. The number of iterations in the For-loop of Lines 10–30 is no more

than $(n[i])^2$. It is obvious that the computational complexity of Algorithm 3 is polynomial.

TABLE III
CONDITIONS FOR ADJUSTING POST-PROCESSING TIME IN A TOOL

$Z_{i-1} > 0$	$Z_i > 0$	$Z_{i+1} > 0$	Operation
False	False	False	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
False	False	True	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
False	True	False	Unchanged
False	True	True	Unchanged
True	False	False	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
True	False	True	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
True	True	False	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
True	True	True	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
×	True	True	Unchanged
×	True	False	Unchanged
×	False	True	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
×	False	False	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
False	False	×	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
False	True	×	Unchanged
True	False	×	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
True	True	×	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$
False	×	×	Unchanged
True	×	×	Increase $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$

C. Post-Processing Time of Two Adjacent Cluster Tools

To keep a high quality of circuit chips on wafers, in real-world production, it is not recommended that the sum of post-processing time $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ in some cluster tools is longer than that in others and in particular, between two adjacent tools, the difference in $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ for them is too large. Consequently, it is necessary to adjust $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ for two adjacent cluster tools.

By Algorithms 1–3, $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ for each tool C_i is minimized. Therefore, $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ cannot be reduced anymore. However, we can appropriately increase the sum of post-processing time of a cluster tool to reduce the difference between $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ and $\sum_{j \in \mathbf{N}_{n[i+1]}^+ \setminus \{b[i+1]\}} r_{i+1,j}$, $i \in \mathbf{N}_{K-1}^+$.

Let $Z_i = \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j} - \sum_{j \in \mathbf{N}_{n[i+1]}^+ \setminus \{b[i+1]\}} r_{i+1,j}$, $i \in \mathbf{N}_{K-1}^+$, denote the difference in total post-processing time between two adjacent cluster tools. Table III determines the cluster tools whose $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$, $i \in \mathbf{N}_K^+$, should be increased.

Reducing the robot waiting time for an individual tool at some processing steps leads to an increase in its robot waiting time at LLs and BMs. As a result, its post-processing time is also increased. During such an adjustment for robot waiting time, (15)–(17) should be satisfied. It is known that Algorithms 2 and 3 determine the post-processing time $r_{i,j}$, which should now be adjusted. After $r_{i,j}$ is adjusted, its new value is

denoted by $r'_{i,j}$. Let $v_i = \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r'_{i,j} - \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ be an increment in post-processing time. Let $v_i \in [0, \gamma_i]$, $i \in \mathbf{N}_K^+$, where γ_i is the upper bound of v_i . To find each minimum v_i , $i \in \mathbf{N}_K^+$, a linear programming model is given as follows.

Linear programming model (LPM):

Maximize γ_i

s.t.

$$\gamma_i \leq \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} \omega_{i,jS}, \quad i \in \mathbf{N}_K^+ \quad (18)$$

$$\gamma_i \leq \tau_{(i+1),0} - (2\lambda_i + \rho_i + \omega_{i,b[i]S}), \quad i = 1 \quad (19)$$

$$\begin{aligned} \gamma_i &\leq (\tau_{(i-1),b[i]} - (2\lambda_i + \rho_i + \omega_{i,0S})) \\ &\quad + (\tau_{(i+1),0} - (2\lambda_i + \rho_i + \omega_{i,b[i]S})) \\ 2 &\leq i \leq K-1 \end{aligned} \quad (20)$$

$$\gamma_i \leq (\tau_{(i-1),b[i]} - (2\lambda_i + \rho_i + \omega_{i,0S})), \quad i = K \quad (21)$$

$$\gamma_i \leq |Z_i|, \quad i = 1 \quad (22)$$

$$\gamma_i \leq |Z_i|, \quad 2 \leq i \leq K-1 \quad (23)$$

$$\gamma_i \leq |Z_{i-1}|, \quad 2 \leq i \leq K-1 \quad (24)$$

$$\gamma_i \leq |Z_{i-1}|, \quad i = K \quad (25)$$

$$0 \leq \gamma_i, \quad 0 \leq i \leq K. \quad (26)$$

The above LPM can get the maximum γ_i . In LPM, (18) indicates that there should be enough robot idle time at the processing steps to be moved to be a part of $r_{i,j}$. Inequalities in (18) make sure that a part of robot idle time can be moved from $\omega_{i,jS}$'s ($j \in \mathbf{N}_{n[i]} \setminus \{0, b[i]\}$) to $\omega_{i,j}$'s ($j \in \{0, b[i]\}$).

Inequalities (19)–(21) indicate that the increased post-processing time should satisfy (16) and (17). By (17), we have $\tau_{(i+1),0} \geq 2\lambda_i + \rho_i + \omega_{i,b[i]S}$. While adjusting the waiting time of R_1 , we need to make (17) hold, leading to that $\tau_{2,0} \geq (2\lambda_1 + \rho_1 + \omega_{1,b[1]S})$ holds. Note that λ_1 and ρ_1 are constant, and $\tau_{2,0}$ is unchanged when a schedule is given by Algorithm 1. Let χ denote the increment in $\omega_{1,b[1]S}$. Then, we have $\tau_{2,0} \geq (2\lambda_1 + \rho_1 + \omega_{1,b[1]S} + \chi)$. Thus, $\tau_{2,0} - (2\lambda_1 + \rho_1 + \omega_{1,b[1]S}) \geq \chi = \gamma_1$. Similarly, we can get (21) for C_K .

While adjusting the waiting time of R_i , $2 \leq i \leq K-1$, we should make (16) and (17) hold for the adjacent cluster tools. Thus, $\tau_{(i-1),b[i]} \geq (2\lambda_i + \rho_i + \omega_{i,0S})$ and $\tau_{(i+1),0} \geq (2\lambda_i + \rho_i + \omega_{i,b[i]S})$ hold. λ_i and ρ_i are constants, and $\tau_{(i-1),b[i]}$ and $\tau_{(i+1),0}$ are unchanged when a schedule is given by Algorithm 1. Let χ_1 and χ_2 denote the increments in $\omega_{i,0S}$ and $\omega_{i,b[i]S}$, respectively. Then, we have $\tau_{(i-1),b[i]} \geq (2\lambda_i + \rho_i + \omega_{i,0S} + \chi_1)$ and $\tau_{(i+1),0} \geq (2\lambda_i + \rho_i + \omega_{i,b[i]S} + \chi_2)$. Thus, $(\tau_{(i-1),b[i]} - (2\lambda_i + \rho_i + \omega_{i,0S})) + (\tau_{(i+1),0} - (2\lambda_i + \rho_i + \omega_{i,b[i]S})) \geq \chi_1 + \chi_2 = \gamma_i$.

Inequalities (22)–(25) mean that the increased post-processing time cannot exceed the difference between the adjacent cluster tools. Obviously, the increased post-processing time should satisfy (26) and an efficient adjustment on post-processing time ranges from 0 to γ_i . After γ_i is determined by LPM, the increment in post-processing time $v_i \in [0, \gamma_i]$ can be given according to a real-world production scenario.

Algorithm 4. Adjusting Post-Processing Time for C_i

Input: v_i
Output: $\omega_{i,j}, \omega_{i,jS}, r_{i,j}$

```

1:  $V \leftarrow \mathbf{N}_{n[i]} \setminus \{0, b[i]\}, M \leftarrow \emptyset$ 
2: For  $j \in V$  do
3:   If  $\Pi_{i,jU} < \Theta$  Then  $M \leftarrow M \cup \{j\}$ 
4:   EndIf
5: EndFor
6:  $\omega_{i,j} \leftarrow \omega_{i,j} + v_i / |V|$  for  $j \in V$ 
7: While  $V \neq \emptyset$  Then
8:    $\Delta \leftarrow v_i / |V|, A \leftarrow \emptyset, B \leftarrow \emptyset$ 
9:   For  $j \in V$  do
10:    If  $j \in M$  Then
11:      If  $\Delta \leq \Pi_{i,jU} - \Pi_{i,jL} - r_{i,j}$  then  $A \leftarrow A \cup \{j\}$ 
12:      Else  $B \leftarrow B \cup \{j\}$ 
13:      EndIf
14:    Else
15:      If  $\Delta \leq \omega_{i,jS}$  then  $A \leftarrow A \cup \{j\}$ 
16:      Else  $B \leftarrow B \cup \{j\}$ 
17:      EndIf
18:    EndIf
19:   EndFor
20:   If  $B \neq \emptyset$  Then
21:     For each  $j \in B$  do
22:       If  $j \in M$  Then
23:          $v_i \leftarrow v_i - (\Pi_{i,jU} - \Pi_{i,jL} - r_{i,j}), r_{i,j} \leftarrow \Pi_{i,jU} - \Pi_{i,jL}, \omega_{i,jS} \leftarrow \Theta - \Pi_{i,jU}$ 
24:       Else
25:          $v_i \leftarrow v_i - \omega_{i,jS}, r_{i,j} \leftarrow \Theta - \Pi_{i,jL}, \omega_{i,jS} \leftarrow 0$ 
26:       EndIf
27:     EndFor
28:      $V \leftarrow V \setminus B$ 
29:   Else
30:     For each  $j \in A$  do
31:        $r_{i,j} \leftarrow \Delta + r_{i,j}, \omega_{i,jS} \leftarrow \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) - r_{i,j}$ 
32:     EndFor
33:      $V \leftarrow \emptyset$ 
34:   EndIf
35: EndWhile
36: End

```

Table III determines a set of tools, denoted by C_{-A} in which post-processing time should be adjusted. LPM gives the adjusted value v_i for the post-processing time in $C_i \in C_{-A}$. Then, Algorithm 4 calculates the post-processing time adjustment for $C_i \in C_{-A}$.

Theorem 3: For $\forall v_i \in [0, \gamma_i]$, Algorithm 4 can find an efficient schedule, which balances the post-processing time between a pair of adjacent cluster tools.

Proof: By Table III, we determine the robots' waiting time that should be adjusted. By LPM, we find the maximum value γ_i that is set for a certain cluster tool. v_i ranges from 0 to γ_i , and obviously a schedule can still be obtained. Then, when we balance the post-processing time of adjacent tools, we ensure that the difference in post-processing time at each processing step in the cluster tool remains unchanged.

By Lines 1 and 3 of Algorithm 4, we have $V = \{j \mid j \neq 0 \text{ and } j \neq b[i], j \in \mathbf{N}_{n[i]}\}$ and $M = \{j \mid \Pi_{i,jU} < \Theta, j \in V\}$. For $j \in V$, we

set $\omega_{i,j} = \omega_{i,j} + v_i / |V|$ in Line 6. Then, Lines 7–35 present the specific adjustment operations for each processing module. By Lines 8, 23 and 31, $\Delta = v_i / |V|$ means that the post-processing time at a processing step is increased.

For $j \in V$, when $j \in M$, if Step j satisfies $\Delta \leq \Pi_{i,jU} - \Pi_{i,jL} - r_{i,j}$, it is put into set A , otherwise, it is put into set B . Lines 15 and 16 indicate that when $j \in V \setminus M$, if Step j satisfies $\Delta \leq \omega_{i,jS}$, it is put into set A , otherwise, it is put into set B . If $B \neq \emptyset$, Lines 20–28 present the adjustment of post-processing time for step $j \in B$ as follows:

1) If $j \in M$, we have $v_i = v_i - (\Pi_{i,jU} - \Pi_{i,jL} - r_{i,j}), r_{i,j} = \Pi_{i,jU} - \Pi_{i,jL}$, and $\omega_{i,jS} = \Theta - \Pi_{i,jU}$, and

2) if $j \in B \setminus M$, we have $v_i = v_i - \omega_{i,jS}, r_{i,j} = \Theta - \Pi_{i,jL}, \omega_{i,jS} = 0$. Lines 30–33 present the adjustment of post-processing time for step $j \in A$, i.e., when $B = \emptyset$, we have $r_{i,j} = \Delta + r_{i,j}, \omega_{i,jS} = \Theta - (\alpha_{i,j} + 2\lambda_i + \rho_i) - r_{i,j}$.

So far, we have carried out the adjustment of post-processing time for tool C_i . The wafer post-processing time in C_i increases so that $|Z_{i-1}|$ and $|Z_i|$ become smaller. $Z_i = \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j} - \sum_{j \in \mathbf{N}_{n[i+1]}^+ \setminus \{b[i+1]\}} r_{i+1,j}$, $i \in \mathbf{N}_{K-1}^+$, means the difference in post-processing time between two adjacent cluster tools. The smaller the value of $|Z_i|$ is, the smaller the post-processing time difference between two adjacent cluster tools becomes. Thus, the wafer post-processing time in adjacent cluster tools is balanced as well as possible. ■

Algorithm 4 makes calculations based on closed-form expressions. The number of iterations in the For-loop of Lines 2–5 and Line 6 is no more than $n[i]$. The number of iterations in the For-loop of Lines 7–35 is no more than $(n[i])^2$. Thus, the computational complexity of Algorithm 4 is polynomial.

D. Algorithm Analysis

Algorithm 1 decides whether a K -cluster tool is schedulable; if so, then Algorithm 1 checks the case that each tool belongs to, and calls Algorithms 2 or 3 to calculate the robot waiting time. By doing so, a schedule can be found to reach the optimal cycle time of a K -cluster tool and minimize the total post-processing time. Upon such an initial schedule solution, Table III determines the robot waiting time to be adjusted for cluster tools. LPM, Table III, and Algorithm 4 can well balance the post-processing time difference between adjacent cluster tools.

As aforementioned, the computational complexity of Algorithms 2–4 is $O((n[i])^2)$. Let $e = \max_{i \in \mathbf{N}_K^+} n[i]$. Then, by the “For loop” in Lines 4–19 of Algorithm 1, the number of calculations for $\omega_{i,jS}$ and $\omega_{i,j}$ depends on the total number of processing modules in a K -cluster tool, which is no more than $\sum_{i=1}^K e^2 = e^2 \times K$. Notice that e is a small constant and is no more than six. Overall, the computational complexity of Algorithms 1–4 is polynomial.

IV. EXAMPLES

Example 1: In a four-cluster tool, $\text{PM}_{1,0}$ is LLs. $\text{PM}_{1,0}, \text{PM}_{2,0}, \text{PM}_{3,0}, \text{PM}_{4,0}, \text{PM}_{1,2}, \text{PM}_{2,2}$, and $\text{PM}_{3,2}$ are BMs. The time unit is second and omitted hereafter. For C_1 , we have $(\alpha_{1,0}, \alpha_{1,1},$

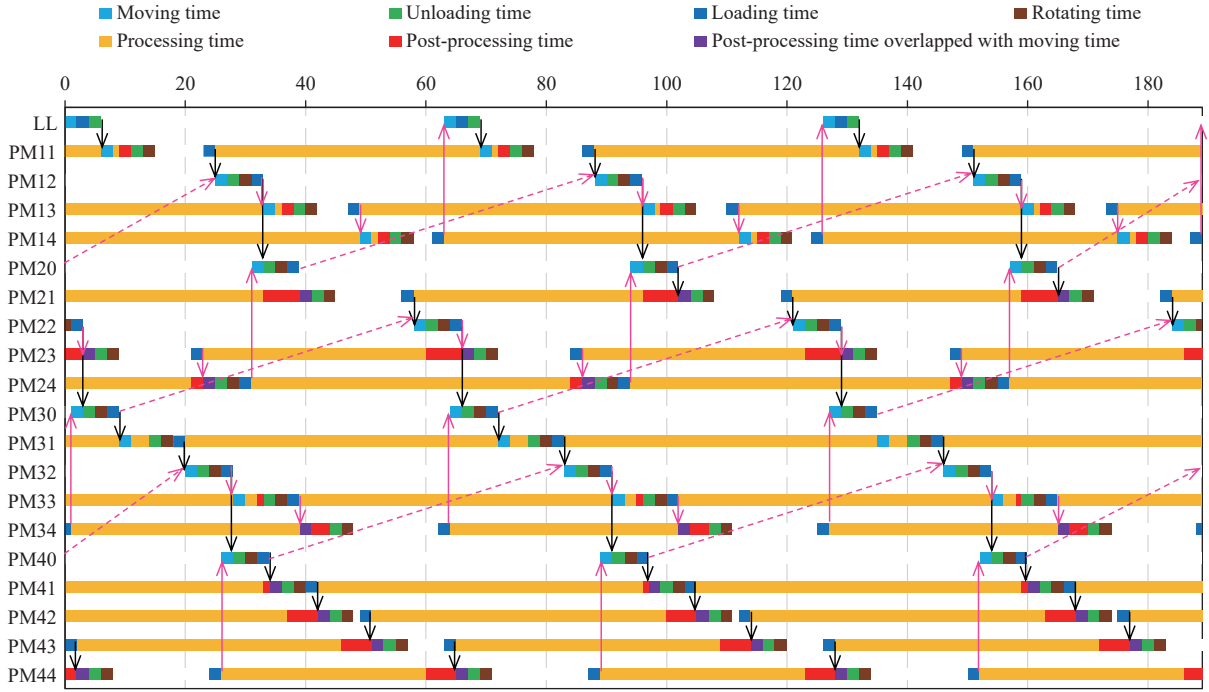


Fig. 2. Gantt chart for the schedule in Example 1 (a four-cluster tool).

$\alpha_{1,2}, \alpha_{1,3}, \alpha_{1,4}; \lambda_1, \mu_1, \rho_1) = (0, 47, 0, 50, 52; 2, 2, 2)$; for C_2 , we have $(\alpha_{2,0}, \alpha_{2,1}, \alpha_{2,2}, \alpha_{2,3}, \alpha_{2,4}; \lambda_2, \mu_2, \rho_2) = (0, 38, 0, 37, 53; 2, 2, 2)$; for C_3 , we have $(\alpha_{3,0}, \alpha_{3,1}, \alpha_{3,2}, \alpha_{3,3}, \alpha_{3,4}; \lambda_3, \mu_3, \rho_3) = (0, 57, 0, 56, 38; 2, 2, 2)$; and for C_4 we have $(\alpha_{4,0}, \alpha_{4,1}, \alpha_{4,2}, \alpha_{4,3}, \alpha_{4,4}; \lambda_4, \mu_4, \rho_4) = (0, 54, 49, 44, 34; 2, 2, 2)$. The wafer processing route is $LLs \rightarrow PM_{1,0} \rightarrow PM_{1,1} \rightarrow PM_{1,2} (PM_{2,0}) \rightarrow PM_{2,1} \rightarrow PM_{2,2} (PM_{3,0}) \rightarrow PM_{3,1} \rightarrow PM_{3,2} (PM_{4,0}) \rightarrow PM_{4,1} \rightarrow PM_{4,2} \rightarrow PM_{4,3} \rightarrow PM_{4,4} \rightarrow PM_{4,0} (PM_{3,2}) \rightarrow PM_{3,3} \rightarrow PM_{3,4} \rightarrow PM_{3,0} (PM_{2,2}) \rightarrow PM_{2,3} \rightarrow PM_{2,4} \rightarrow PM_{2,0} (PM_{1,2}) \rightarrow PM_{1,3} \rightarrow PM_{1,4} \rightarrow LLs$. After being processed, a wafer can stay at $PM_{i,j}$ for no more than $\delta_{1,1} = 20, \delta_{1,3} = 20, \delta_{1,4} = 20, \delta_{2,1} = 20, \delta_{2,3} = 20, \delta_{2,4} = 20, \delta_{3,1} = 20, \delta_{3,3} = 20, \delta_{3,4} = 16, \delta_{4,1} = 20, \delta_{4,2} = 20, \delta_{4,3} = 10$, and $\delta_{4,4} = 13$, respectively. There are no residency time constraints at LLs and BMs.

By (4), (10) and (12), the cycle time of this four-cluster tool is $\Theta = 63$. By Algorithm 1, we have

1) For C_1 , $\psi_{1,1} \leq \Theta$ and $\Theta \leq \Pi_{1,jU}, j \in \mathbb{N}_{n[1]}^+ \setminus \{b[1]\}$, are satisfied and Algorithm 2 is applied;

2) For C_2 , $\psi_{2,1} \leq \Theta$ and $\Theta \leq \Pi_{2,jU}, j \in \mathbb{N}_{n[2]}^+ \setminus \{b[2]\}$, are satisfied and Algorithm 2 is also applied;

3) For C_3 , $[\Pi_{3,1L}, \Pi_{3,1U}] \cap [\Pi_{3,3L}, \Pi_{3,3U}] \cap [\Pi_{3,4L}, \Pi_{3,4U}] = \emptyset$ is satisfied and Algorithm 3 is applied; and

4) For C_4 , $[\Pi_{4,1L}, \Pi_{4,1U}] \cap [\Pi_{4,2L}, \Pi_{4,2U}] \cap [\Pi_{4,3L}, \Pi_{4,3U}] \cap [\Pi_{4,4L}, \Pi_{4,4U}] = \emptyset$ is satisfied and Algorithm 3 is applied.

For C_1 , by Algorithm 2, $\varepsilon = 22$ satisfies $\varepsilon < \psi_{1,2} = 25$. Then, we have $(\omega_{1,0S}, \omega_{1,1S}, \omega_{1,2S}, \omega_{1,3S}, \omega_{1,4S}) = (0, 10, 0, 7, 5)$, $(\omega_{1,0}, \omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \omega_{1,4}) = (0, 1, 0, 1, 1)$, and $(r_{1,1}, r_{1,3}, r_{1,4}) = (0, 0, 0)$.

For C_2 , by Algorithm 2, $\varepsilon = 43$ satisfies $\psi_{2,2} = 23 < \varepsilon$. Obviously, by Δ and $\Theta - (\alpha_{2,j} + 2\lambda_2 + 3\rho_2)$ for $j \in \{1, 3, 4\}$ we have $A = \{1, 3\}$ and $B = \{4\}$. Let $r_{2,4} = \Theta - (\alpha_{2,4} + 3\lambda_2 + \rho_2)$. Then, $V = V \setminus B$, we have $(\omega_{2,0S}, \omega_{2,1S}, \omega_{2,2S}, \omega_{2,3S}, \omega_{2,4S}) = (0, 11, 0, 12, 0)$, $(\omega_{2,0}, \omega_{2,1}, \omega_{2,2}, \omega_{2,3}, \omega_{2,4}) = (0, 0, 0, 0, 0)$, and $(r_{2,1},$

$r_{2,3}, r_{2,4}) = (8, 8, 4)$.

For C_3 , by Algorithm 3, $E = \{4\}$ and $F = \{1, 3\}$, and $V = \{1, 3, 4\}$. By Lines 3 and 4, the waiting time can be set as $(\omega_{3,1S}, \omega_{3,3S}, \omega_{3,4S}) = (0, 0, 3)$. Then, we have $\varepsilon = 20 > \eta = 17$. Thus, we have $(\omega_{3,0S}, \omega_{3,1S}, \omega_{3,2S}, \omega_{3,3S}, \omega_{3,4S}) = (0, 0, 0, 1, 19)$, $(\omega_{3,0}, \omega_{3,1}, \omega_{3,2}, \omega_{3,3}, \omega_{3,4}) = (0, 1, 0, 1, 1)$, and $(r_{3,1}, r_{3,3}, r_{3,4}) = (0, 0, 0)$.

For C_4 , by Algorithm 3, $E = \{3, 4\}$ and $F = \{1, 2\}$, and $V = \{1, 2, 3, 4\}$. By Lines 3 and 4, the waiting time can be set as $(\omega_{4,1S}, \omega_{4,2S}, \omega_{4,3S}, \omega_{4,4S}) = (0, 0, 3, 10)$. Then, we have $\varepsilon = 10 < \eta = 34$. Obviously, by Δ and $\Theta - (\alpha_{4,j} + 2\lambda_4 + \rho_4)$ for $j \in \{1, 2, 3, 4\}$, we have $A = \{2, 3, 4\}$ and $B = \{1\}$. Let $r_{4,j} = \Theta - (\alpha_{4,j} + 4\lambda_4 + \rho_4)$ for $j \in B$. Then, $V = V \setminus B$, we have $(\omega_{4,0S}, \omega_{4,1S}, \omega_{4,2S}, \omega_{4,3S}, \omega_{4,4S}) = (0, 0, 1, 6, 16)$, $(\omega_{4,0}, \omega_{4,1}, \omega_{4,2}, \omega_{4,3}, \omega_{4,4}) = (0, 0, 0, 0, 0)$, and $(r_{4,1}, r_{4,2}, r_{4,3}, r_{4,4}) = (3, 7, 7, 7)$.

By Table II, we know that the post-processing time should be adjusted for C_1 and C_3 . By LPM, we have $\gamma_1 = 20$ and $\gamma_3 = 20$. Then, we have $v_1 \in [0, 20]$ and $v_3 \in [0, 20]$. Assume that the post-processing time difference between adjacent tools of this multi-cluster tool cannot exceed 16. Then, we let $v_1 = 6$ and $v_3 = 6$. By Algorithm 4, $(\omega_{1,0S}, \omega_{1,1S}, \omega_{1,2S}, \omega_{1,3S}, \omega_{1,4S}) = (0, 8, 0, 5, 3)$, $(\omega_{1,0}, \omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \omega_{1,4}) = (0, 3, 0, 3, 3)$, $(r_{1,1}, r_{1,3}, r_{1,4}) = (2, 2, 2)$, $(\omega_{3,0S}, \omega_{3,1S}, \omega_{3,2S}, \omega_{3,3S}, \omega_{3,4S}) = (0, 0, 0, 0, 14)$, $(\omega_{3,0}, \omega_{3,1}, \omega_{3,2}, \omega_{3,3}, \omega_{3,4}) = (0, 3, 0, 3, 3)$, and $(r_{3,1}, r_{3,3}, r_{3,4}) = (2, 2, 2)$. It can be found that $\tau_{i,j} \in [\alpha_{i,j}, \alpha_{i,j} + \delta_{i,j}]$, $i \in \mathbb{N}_K^+$ holds.

After the post-processing time is adjusted, the Gantt chart for this schedule is shown in Fig. 2. The horizontal axis represents time and each horizontal lane marks the activities performed at a PM, which is drawn in different colors. Vertical axis represents different PMs. In each horizontal lane of a PM, let yellow stand for a wafer being processed, blue for a robot's moving, red for a wafer that is waiting to be unloaded after its

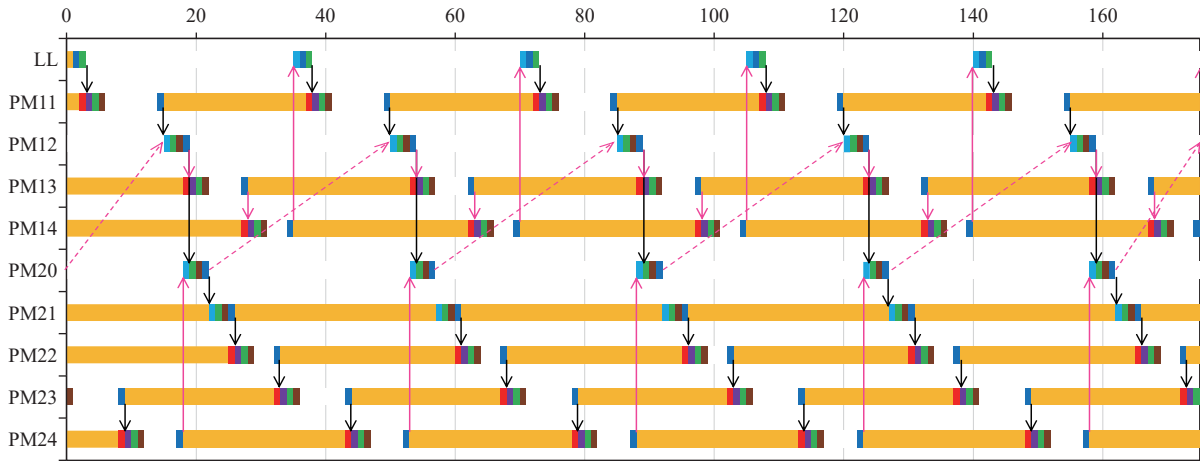


Fig. 3. Gantt chart for the schedule in Example 2 (a two-cluster tool).

processing, green for a robot's unloading a processed wafer, brown for a robot's rotation, a white bar for a PM being idle, and dark blue for a robot's loading a wafer.

Fig. 2 shows that the post-processing time marked with red is short and the differences among the post-processing time marked by red bars are small. In Fig. 2, we use arrows to illustrate the operation routes of robots.

The algorithms in [24] give a schedule by setting the following robot waiting time:

$$\begin{aligned}
 (\omega'_{1,0S}, \omega'_{1,1S}, \omega'_{1,2S}, \omega'_{1,3S}, \omega'_{1,4S}) &= (0, 0, 0, 0, 0) \\
 (\omega'_{1,0}, \omega'_{1,1}, \omega'_{1,2}, \omega'_{1,3}, \omega'_{1,4}) &= (0, 0, 0, 0, 25) \\
 (r'_{1,1}, r'_{1,3}, r'_{1,4}) &= (10, 7, 5) \\
 (\omega'_{2,0S}, \omega'_{2,1S}, \omega'_{2,2S}, \omega'_{2,3S}, \omega'_{2,4S}) &= (0, 0, 0, 0, 0) \\
 (\omega'_{2,0}, \omega'_{2,1}, \omega'_{2,2}, \omega'_{2,3}, \omega'_{2,4}) &= (0, 0, 0, 0, 23) \\
 (r'_{2,1}, r'_{2,3}, r'_{2,4}) &= (19, 20, 4) \\
 (\omega'_{3,0S}, \omega'_{3,1S}, \omega'_{3,2S}, \omega'_{3,3S}, \omega'_{3,4S}) &= (0, 0, 0, 0, 3) \\
 (\omega'_{3,0}, \omega'_{3,1}, \omega'_{3,2}, \omega'_{3,3}, \omega'_{3,4}) &= (0, 0, 0, 0, 20) \\
 (r'_{3,1}, r'_{3,3}, r'_{3,4}) &= (0, 1, 16) \\
 (\omega'_{4,0S}, \omega'_{4,1S}, \omega'_{4,2S}, \omega'_{4,3S}, \omega'_{4,4S}) &= (0, 0, 0, 3, 10) \\
 (\omega'_{4,0}, \omega'_{4,1}, \omega'_{4,2}, \omega'_{4,3}, \omega'_{4,4}) &= (0, 0, 0, 0, 10) \\
 (r'_{4,1}, r'_{4,2}, r'_{4,3}, r'_{4,4}) &= (3, 8, 10, 13).
 \end{aligned}$$

Let $r_{i,j}^{\max}$ and $r_{i,j}^{\min}$ denote the maximum $r_{i,j}$ and minimum $r_{i,j}$ in C_i , respectively. Obviously, we can get

$\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j} < \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r'_{i,j}$, $i \in \mathbf{N}_4^+$, and $\sum_{i=1}^4 \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j} = 56$, $\sum_{i=1}^4 \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r'_{i,j} = 116$. Thus, the total post-processing time of this four-cluster tool is decreased by 51.7%. Furthermore, we have the following results.

- 1) $|r_{i,j}^{\max} - r_{i,j}^{\min}| < |r'_{i,j}^{\max} - r'_{i,j}^{\min}|$, $i \in \mathbf{N}_4^+$, $j \neq f$, $\{j, f\} \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}$; and
- 2) $|Z_i| < |Z'_i|$, $i \in \mathbf{N}_4^+$.

Example 2: In a two-cluster tool, $\text{PM}_{1,0}$ is LLs. $\text{PM}_{2,0}$ and $\text{PM}_{1,2}$ are BMs. The time unit is second and omitted hereafter. For C_1 , we have $(\alpha_{1,0}, \alpha_{1,1}, \alpha_{1,2}, \alpha_{1,3}, \alpha_{1,4}; \lambda_1, \mu_1, \rho_1) = (0, 22, 0, 25, 27; 1, 1, 1)$; for C_2 , we have $(\alpha_{2,0}, \alpha_{2,1}, \alpha_{2,2}, \alpha_{2,3}, \alpha_{2,4}; \lambda_2, \mu_2, \rho_2) = (0, 32, 27, 23, 25; 1, 1, 1)$. The wafer processing route is $\text{LLs} \rightarrow \text{PM}_{1,0} \rightarrow \text{PM}_{1,1} \rightarrow \text{PM}_{1,2} (\text{PM}_{2,0}) \rightarrow \text{PM}_{2,1} \rightarrow \text{PM}_{2,2} \rightarrow \text{PM}_{2,3} \rightarrow \text{PM}_{2,4} \rightarrow \text{PM}_{2,0} (\text{PM}_{1,2}) \rightarrow \text{PM}_{1,3} \rightarrow \text{PM}_{1,4} \rightarrow \text{LLs}$. After being processed, a wafer can stay at $\text{PM}_{i,j}$ for

no more than $\delta_{1,1} = \delta_{1,3} = \delta_{1,4} = \delta_{2,1} = \delta_{2,2} = \delta_{2,3} = \delta_{2,4} = 10$, respectively. There are no wafer residency time constraints at LLs and BMs.

By (4), (10) and (12), the cycle time of this two-cluster tool is $\Theta = 35$. By Algorithm 1, we have

1) For C_1 , $\psi_{1,1} \leq \Theta$ and $\Theta \leq \Pi_{1,jU}$, $j \in \mathbf{N}_{n[1]}^+ \setminus \{b[1]\}$, are satisfied and Algorithm 2 is applied;

2) For C_2 , $\psi_{2,1} \leq \Theta$ and $\Theta \leq \Pi_{2,jU}$, $j \in \mathbf{N}_{n[2]}^+$, are satisfied and Algorithm 2 is also applied.

For C_1 , by Algorithm 2, $\varepsilon = 22$ satisfies $\varepsilon > \psi_{1,2} = 19$. Then, we have $(\omega_{1,0S}, \omega_{1,1S}, \omega_{1,2S}, \omega_{1,3S}, \omega_{1,4S}) = (0, 8, 0, 5, 3)$, $(\omega_{1,0}, \omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \omega_{1,4}) = (0, 0, 0, 0, 0)$, and $(r_{1,1}, r_{1,3}, r_{1,4}) = (2, 2, 2)$.

For C_2 , by Algorithm 2, $\varepsilon = 21$ satisfies $\varepsilon > \psi_{2,2} = 23$. Then, we have $(\omega_{2,0S}, \omega_{2,1S}, \omega_{2,2S}, \omega_{2,3S}, \omega_{2,4S}) = (0, 0, 3, 7, 5)$, $(\omega_{2,0}, \omega_{2,1}, \omega_{2,2}, \omega_{2,3}, \omega_{2,4}) = (0, 0, 0, 0, 0)$, and $(r_{2,1}, r_{2,2}, r_{2,3}, r_{2,4}) = (0, 2, 2, 2)$.

Fig. 3 shows that the post-processing time marked with red is short and the post-processing time difference between the adjacent tools marked by red bars is small. In Fig. 3, we use arrows to illustrate the operation routes of robots.

The algorithms in [24] give a schedule by setting the following robot waiting time:

$$\begin{aligned}
 (\omega'_{1,0S}, \omega'_{1,1S}, \omega'_{1,2S}, \omega'_{1,3S}, \omega'_{1,4S}) &= (0, 0, 0, 0, 0) \\
 (\omega'_{1,0}, \omega'_{1,1}, \omega'_{1,2}, \omega'_{1,3}, \omega'_{1,4}) &= (0, 0, 0, 0, 16) \\
 (r'_{1,1}, r'_{1,3}, r'_{1,4}) &= (10, 7, 5) \\
 (\omega'_{2,0S}, \omega'_{2,1S}, \omega'_{2,2S}, \omega'_{2,3S}, \omega'_{2,4S}) &= (0, 0, 0, 0, 0) \\
 (\omega'_{2,0}, \omega'_{2,1}, \omega'_{2,2}, \omega'_{2,3}, \omega'_{2,4}) &= (0, 0, 0, 0, 15) \\
 (r'_{2,1}, r'_{2,3}, r'_{2,4}) &= (0, 5, 9, 7).
 \end{aligned}$$

Obviously, we can get

$$\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j} < \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r'_{i,j}, i \in \mathbf{N}_2^+$$

$\sum_{i=1}^2 \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j} = 12$, $\sum_{i=1}^2 \sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r'_{i,j} = 43$. Thus, the total post-processing time of this four-cluster tool is decreased by 72.0%. Furthermore, we have the following results.

- 1) $|r_{i,j}^{\max} - r_{i,j}^{\min}| < |r'_{i,j}^{\max} - r'_{i,j}^{\min}|$, $i \in \mathbf{N}_2^+$, $j \neq f$, $\{j, f\} \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}$; and
- 2) $|Z_i| < |Z'_i|$, $i \in \mathbf{N}_2^+$.

The results of the above examples have been shown as bar

graphs in Figs. 4 and 5. It is obvious that the total post-processing time and the post-processing time difference between different steps are significantly less for C_i than that for C'_i . Compared with the schedule obtained by [24], the obtained schedule by the proposed method achieves the following goals.

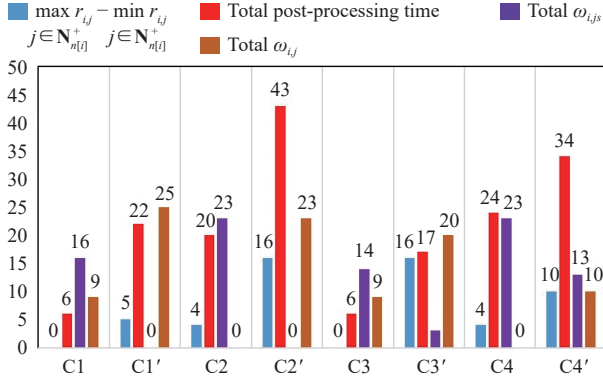


Fig. 4. Comparison for Example 1.

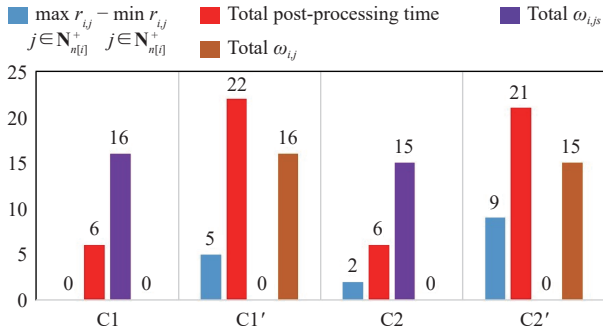


Fig. 5. Comparison for Example 2.

- 1) Minimize $\sum_{j \in \mathbf{N}_{n[i]}^+ \setminus \{b[i]\}} r_{i,j}$ for C_i , $i \in \mathbf{N}_K^+$;
- 2) Balance the difference in post-processing time among $\text{PS}_{i,j}$, $j \in \mathbf{N}_{n[i]} \setminus \{0, b[i]\}$; and
- 3) Balance the post-processing time difference between adjacent cluster tools.

V. CONCLUSION

Multi-cluster tools are widely used in wafer manufacturing in the semiconductor manufacturing industry due to their high efficiency. In order to load/unload wafers according to the wafer route, proper cooperation between the robots of each cluster tool is demanded. Furthermore, after a wafer is processed in a processing module, the robot is scheduled to unload it timely because the decreasing narrow circuit width demands limit the post-processing time. This study finds an optimal schedule for a multi-cluster tool to minimize the wafer post-processing time. Besides, we make efforts to avoid uneven post-processing time among the processing steps. The proposed method can handle the case where the difference in post-processing time between two adjacent cluster tools is large. The method can greatly lead to high-yield and high-quality integrated-circuit chips on a wafer.

In future work, we will investigate how to regulate the

wafer post-processing time for a multi-cluster tool when its processing time is disturbed.

REFERENCES

- [1] 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. Semiconductor Manufacturing*, vol. 10, no. 4, pp. 418–424, Nov. 1997.
- [2] Y.-H. Shin, T.-E. Lee, J.-H. Kim, and H.-Y. Lee, "Modeling and implementing a real-time scheduler for dual-armed cluster tools," *Computers Industry*, vol. 45, no. 1, pp. 13–27, May 2001.
- [3] M. Lee, J. R. Morrison, and A. A. Kalir, "Practical queueing models for preventive maintenance plan optimization: Multiple maintenance types and numerical studies," *IEEE Trans. Semiconductor Manufacturing*, vol. 34, no. 1, pp. 104–114, Feb. 2021.
- [4] 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.
- [5] 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. Semiconductor Manufacturing*, vol. 21, no. 2, pp. 224–237, May 2008.
- [6] 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. Systems, Man, and Cybernetics: Systems*, vol. 43, no. 1, pp. 196–207, Jan. 2013.
- [7] 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. Automation Science and Engineering*, vol. 5, no. 3, pp. 490–503, Jul. 2008.
- [8] H. J. Yoon and D. Y. Lee, "Online scheduling of integrated single-wafer processing tools with temporal constraints," *IEEE Trans. Semiconductor Manufacturing*, vol. 18, no. 4, pp. 390–398, Nov. 2005.
- [9] C. Sriskandarajah, I. Drobouchevitch, and S. P. Sethi, "Scheduling multiple parts in a robotic cell served by a dual-gripper robot," *Operations Research*, vol. 52, no. 1, pp. 65–82, Feb. 2004.
- [10] W. M. Zuberek, "Timed petri nets in modeling and analysis of cluster tools," *IEEE Trans. Robotics and Automation*, vol. 17, no. 5, pp. 562–575, Oct. 2001.
- [11] M. Meyyappan, "A review of plasma enhanced chemical vapour deposition of carbon nanotubes," *J. Physics D: Applied Physics*, vol. 42, no. 21, pp. 213001–213015, Oct. 2009.
- [12] W. Liu, S. Kraemer, D. Sarkar, P. M. Ajayan, and K. Banerjee, "Controllable and rapid synthesis of high-quality and large-area Bernal stacked bilayer graphene using chemical vapor deposition," *Chemistry Materials*, vol. 26, no. 2, pp. 907–915, Jan. 2014.
- [13] J.-H. Lee and H.-J. Kim, "Completion time analysis of wafer lots in single-armed cluster tools with parallel processing modules," *IEEE Trans. Autom. Science Engineering*, vol. 14, no. 4, pp. 1622–1633, Oct. 2017.
- [14] 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. Autom. Science Engineering*, vol. 16, no. 2, pp. 898–907, Apr. 2019.
- [15] D.-H. Roh, T.-G. Lee, and T.-E. Lee, "K-cyclic schedules and the worst-case wafer delay in a dual-armed cluster tool," *IEEE Trans. Semiconductor Manufacturing*, vol. 32, no. 2, pp. 236–249, May 2019.
- [16] J.-H. Lee, H.-J. Kim, and T.-E. Lee, "Scheduling cluster tools for concurrent processing of two wafer types," *IEEE Trans. Autom. Science Engineering*, vol. 11, no. 2, pp. 525–536, Apr. 2014.
- [17] Y. Qiao, N. Q. Wu, F. J. Yang, M. C. Zhou, Q. H. Zhu, and T. Qu, "Robust scheduling of time-constrained dual-arm cluster tools with wafer revisiting and activity time disturbance," *IEEE Trans. Systems,*

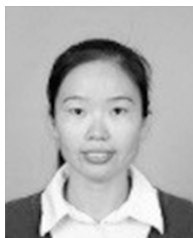
- Man, Cybernetics: Systems*, vol. 49, no. 6, pp. 1228–1240, Jun. 2019.
- [18] F. J. Yang, N. Q. Wu, Y. Qiao, and R. Su, “Polynomial approach to optimal one-wafer cyclic scheduling of treelike hybrid multi-cluster tools via Petri nets,” *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 270–280, Jan. 2018.
 - [19] 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. Systems, Man, Cybernetics: Systems*, vol. 48, no. 4, pp. 622–636, Apr. 2018.
 - [20] T. Nishi, Y. Watanabe, and M. Sakai, “An efficient deadlock prevention policy for noncyclic scheduling of multicluster tools,” *IEEE Trans. Autom. Science Engineering*, vol. 15, no. 4, pp. 1677–1691, Oct. 2018.
 - [21] 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. Production Research*, vol. 55, no. 11, pp. 3092–3109, Jun. 2017.
 - [22] 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.
 - [23] 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. Systems, Man, Cybernetics: Systems*, vol. 46, no. 10, pp. 1456–1467, Oct. 2016.
 - [24] 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.
 - [25] 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. Autom. Science Engineering*, vol. 12, no. 2, pp. 642–655, Apr. 2015.
 - [26] T. Nishi and I. Matsumoto, “Petri net decomposition approach to deadlock-free and non-cyclic scheduling of dual-armed cluster tools,” *IEEE Trans. Autom. Science Engineering*, vol. 12, no. 1, pp. 281–294, Jan. 2015.
 - [27] W. K. V. Chan, J. G. Yi, and S. W. Ding, “Optimal scheduling of multicluster tools with constant robot moving times, Part I: Two-cluster analysis,” *IEEE Trans. Autom. Science Engineering*, vol. 8, no. 1, pp. 5–16, Jan. 2011.
 - [28] 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, Sept. 2004.
 - [29] M. Dawande, C. Srikandarajah, and S. Sethi, “On throughput maximization in constant travel-time robotic cells,” *Manufacture and Service Operation Manage.*, vol. 4, no. 4, pp. 296–312, Oct. 2002.
 - [30] D. Jevtic, “Method and apparatus for managing scheduling a multiple cluster tool,” European Patent 1 132 792(A2), Mar. 3, 2001.
 - [31] Z. Cao, C. Lin, M. Zhou, and R. Huang, “Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling,” *IEEE Trans. Autom. Science Engineering*, vol. 16, no. 2, pp. 825–837, Apr. 2019.
 - [32] 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. Systems, Man, & Cybernetics: Systems*, vol. 48, no. 3, pp. 389–400, Mar. 2018.
 - [33] J. Luo, Z. Liu, M. Zhou, and K. Xing, “Deadlock-free scheduling of flexible assembly systems based on Petri nets and local search,” *IEEE Trans. Systems, Man, Cybernetics: Systems*, vol. 50, no. 10, pp. 3658–3669, Oct. 2020.
 - [34] J. Luo, M. Zhou, and J. Q. Wang, “AB&B: An anytime branch and bound algorithm for scheduling of deadlock-prone flexible manufacturing systems,” *IEEE Trans. Autom. Science Engineering*, vol. 18, no. 4, pp. 2011–2021, Oct. 2021.
 - [35] B. Huang, and M. Zhou, “Symbolic scheduling of robotic cellular manufacturing systems with timed Petri nets,” *IEEE Trans. Control Systems Technology*, vol. 30, no. 5, pp. 1876–1887, Sept. 2022.
 - [36] S. Wang, X. Guo, O. Karoui, M. Zhou, D. You, and A. Abusorrah, “A refined siphon-based deadlock prevention policy for a class of Petri nets,” *IEEE Trans. Systems, Man, Cybernetics: Systems*, vol. 53, no. 1, pp. 191–203, Jan. 2023.
 - [37] Q. Zhu, N. Wu, Y. Qiao, and M. Zhou, “Petri net-based optimal one-wafer scheduling of single-arm multi-cluster tools in semiconductor manufacturing,” *IEEE Trans. Semiconductor Manufacturing*, vol. 26, no. 4, pp. 578–591, Feb. 2013.
 - [38] 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. Autom. Science Engineering*, vol. 5, no. 2, pp. 321–336, Apr. 2008.
 - [39] N. Q. Wu, and M. C. Zhou, “Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets,” *IEEE Trans. Autom. Science Engineering*, vol. 9, no. 2, pp. 446–454, Apr. 2012.
 - [40] N. Q. Wu and M. C. Zhou, “Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation,” *IEEE Trans. Autom. Science Engineering*, vol. 9, no. 1, pp. 203–209, Jan. 2012.
 - [41] N. Q. Wu and M. C. Zhou, “A closed-form solution for schedulability and optimal scheduling of dual-arm cluster tools with wafer residency time constraint based on steady schedule analysis,” *IEEE Trans. Autom. Science Engineering*, vol. 7, no. 2, pp. 303–315, Apr. 2010.
 - [42] Y. Y. Yan, H. G. Wang, Q. H. Tao, W. H. Fan, T. Y. Lin, and Y. Y. Xiao, “Noncyclic scheduling of multi-cluster tools with residency constraints based on pareto optimization,” *IEEE Trans. Semiconductor Manufacturing*, vol. 33, no. 3, pp. 476–486, Aug. 2020.
 - [43] X. Chen, Y. Xie, Y. Sheng, H. Tang, Z. Wang, Y. Wang, Y. Wang, F. Liao, J. Ma, and X. Guo, “Wafer-scale functional circuits based on two dimensional semiconductors with fabrication optimized by machine learning,” *Nature Communications*, vol. 12, no. 1, pp. 1–8, Oct. 2021.
 - [44] Q. Zhu, M. Zhou, Y. Qiao, N. Wu, and Y. Hou, “Multiobjective scheduling of dual-blade robotic cells in wafer fabrication,” *IEEE Trans. Systems, Man, Cybernetics: Systems*, vol. 50, no. 12, pp. 5015–5023, Dec. 2020.



Qinghua Zhu (Senior Member, IEEE) received the Ph.D. degree in industrial engineering from Guangdong University of Technology in 2013. From 2014 to 2015 and from 2017 to 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 intelligent manufacturing scheduling, cloud computing, edge computing, discrete event systems, and Petri nets.



Bin Li received the B.S. degree in software engineering from Jiangxi University of Finance and Economics in 2020. He is currently a master student in computer technology at the School of Computer Science and Technology, Guangdong University of Technology. His research interests include intelligent manufacturing scheduling and discrete event systems.



engineering.

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 the Guangdong University of Technology in 2016. Since 2002, she has been with the School of Computer Science and Technology, Guangdong University of Technology where she is currently an Associate Professor. Her current research interests include production scheduling and optimization, information systems, and software



Hongpeng Li received the B.S. degree in computer science and technology from University of Electronic Science and Technology of China, Zhongshan Institute in 2000. He is currently a master student in computer science and technology at the School of Computer Science and Technology, Guangdong University of Technology. His research interests include intelligent manufacturing scheduling and discrete event systems.



Naiqi Wu (Fellow, IEEE) received the B.S. degree in electrical engineering from Anhui University of Science and 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, China, 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 Chair Professor at the Department of Engineering Science and the Institute of Systems Engineering, Macau University of Science and Technology, China. 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 co-author of one book, five book chapters, and over 200 peer-reviewed journal papers. He 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*.