Scheduling of Multi-Finger-Robotic Cluster Tools with Multi-Space Process Modules

Lei Gu, Naiqi Wu, Yan Qiao and Siwei Zhang

October 28, 2024

Scheduling of Multi-finger-robotic Cluster Tools with Multi-space Process Modules

Lei Gu
Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems Macau University of Science and Technology
Macao, China
gulei0618@163.com

NaiQi Wu
Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems Macau University of Science and Technology
Macao, China
nqwu@must.edu.mo

Yan Qiao
Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems Macau University of Science and Technology
Macao, China
yqiao@must.edu.mo

SiWei Zhang
Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems Macau University of Science and Technology
Macao, China
swzhang@must.edu.mo

***Abstract*—Robotic cluster tools have been applied for wafer fabrication. To improve the productivity of such a tool, one method involves increasing the number of spaces within a process module and the fingers on a robot arm, leading to multi-finger-robotic cluster tools with multi-space process modules. This study addresses the scheduling issue of four-finger-robotic cluster tools with four-space process modules, a commonly seen configuration in modern semiconductor fabs. Residency time constraints make the scheduling issue of such a tool more challenging. This study conducts an analysis of the steady-state operations of such a tool under two presented robot task sequences. Then, based on the temporal properties of the tool, two linear programs are developed to find feasible schedules to minimize the system cycle time. Experiments show the performance of a tool under different robot task sequences by considering different parameters.**

***Keywords*—Semiconductor manufacturing, cluster tools, scheduling, linear program**

I. Introduction

Cluster tools have demonstrated adaptability, and reconfigurability for wafer fabrication in semiconductor manufacturing, resulting in enhanced efficiency, shorter cycle times [1], optimized space usage [2], and decreased capital expenses. Cluster tools generally include multiple *process modules* (PMs), a wafer-handling robot, and two *loadlocks* (LLs) for loading and unloading wafer cassettes. Each cassette carries 25 wafers associated with an identical recipe [3]. A cluster tool with single-space PM and single-finger-arm robot is called S^{3}F-CT. Processing wafers in PMs utilizes single-wafer processing technology, which permits each PM to handle only one wafer at a time. A robot transfers a raw wafer from a cassette to PMs in a predetermined sequence and returns it to the cassette once all operations are finished at the PMs.

For some processes, cluster tools have more spaces in PMs and additional fingers on robot arms. As illustrated in Fig. 1, these tools include multiple four-space PMs, a dual-arm robot, and several LLs. Each four-space PM includes two spaces facing the door for transferring wafers in and out and two spaces on the other side. The inner chamber rotates to allow the robot to access all spaces. Each robot arm has two closely positioned fingers, allowing just one arm to load or unload at a time. Thus, one arm with two fingers can access a PM or an LL to load or unload two wafers concurrently. Once four wafers are loaded into a PM, the chamber rotates to implement a first-in-first-out unloading sequence. With four LLs ensuring the timely entry of raw wafers, the robot can perform tasks at an LL without waiting, as LLs do not limit throughput. The cluster tool shown in Fig. 1, featuring four-space PMs and a four-finger-arm robot, is named SF^3-CT.

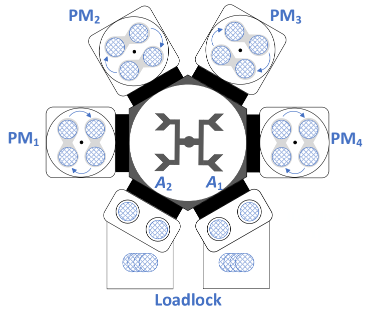

Figure 1. A cluster tool with four-space PMs

As wafer circuits are minimized to nanometer dimensions, preventing impurities becomes essential. To do so, there are *residency time constraints* (RTCs) [4]. By RTCs, wafers need to be removed from a PM within a specified time interval after processed. This prevents contamination or damage from particles, high temperatures, and chemical gases. Proper use of robot fingers is also essential. To avoid cross-contamination, fingers are designated either clean or dirty. Dirty fingers load or unload raw wafers, while clean fingers load or unload processed wafers. In semiconductor fabs, as shown in Fig. 1, one robot arm has two dirty fingers, and the other arm has two clean fingers. SF^3-CTs are mainly used for chemical vapor deposition processes. So far, studies on SF^3-CTs scheduling are insufficient. The majority of current research focus on S^{3}F-CTs [1–22].

Typically, enough raw wafers are introduced for processing to ensure steady operations. In practice, the wafer processing time in a PM is much longer than the robot task time [5]. If the robot task time per cycle exceeds the wafer completion time at a step, the cluster tool becomes transport-bound, keeping the robot continually busy. Conversely, the tool is process-bound, causing some robot idle time. For process-bound tools, a backward strategy is ideal for cluster tools with single-arm robots [6], and a swap strategy is effective for the ones with

This work was supported in part by Science and Technology development fund (FDCT), Macau SAR (file Nos. 0083/2021/A2 and 0011/2023/RIA1).

dual-arm robots [7]. Significant progresses have been made in the simulation, performance analysis, regulation, and planning of cluster tools in the steady state [8–12]. However, RTCs are not considered in those studies.

In [4], [13], and [14], various methods are suggested to determine the best cyclic schedules for dual-arm-robotic cluster tools considering RTCs. Additionally, Wu *et al*. [3] and [15] study the scheduling issues for single-arm and dual-arm cluster tools using control theory to obtain optimal schedules efficiently. Lim *et al*. [16] present novel robot task sequences that better meet RTCs with unbalanced processing conditions. Ko *et al*. [17] propose a method for dual-arm-robotic cluster tools dealing with various wafer types, intending to meet RTCs and shorten cycle time by determining the optimal release sequence. For cluster tools with variations in processing time and RTCs, Lim *et al*. [18] recommend an adaptable approach to improve scheduling resilience. Addressing the scheduling issue of multi-cluster tools with RTCs, Zhu *et al*. [19] propose an efficient steady-state schedule for processing two wafer types simultaneously. Initiatives to simulate, control, and organize reentrant processes with RTCs in single and dual-arm-robotic cluster tools are covered in [20, 21]. [22] explores the effect of activity duration fluctuation on wafer delay in PMs.

For S^{3}F-CTs, wafer processing starts immediately after entering a PM, making it important to analyze the robot tasks [3, 15]. Additionally, PM chambers must rotate for correct wafer loading. If a PM is rotating when the robot arrives, the robot must wait, delaying the following robot tasks. Hence, for SF^3-CTs, evaluating how PM rotations affect the robot task schedule is crucial. Furthermore, results in S^{3}F-CT studies are not applicable to SF^3-CTs directly. RTCs and the requirement to effectively manage clean and dirty fingers bring further complexity to the SF^3-CT scheduling issue. This motivates us to conduct this work. To do so, two scheduling strategies are introduced based on system properties, and two corresponding linear programs are established to find feasible and optimal schedules. Also, experiments are conducted to evaluate these methods' effectiveness.

II. The Processes and Scheduling Strategies

A. The Operation Processes

Wafer flow patterns (WFPs) indicate the processing paths and WFP = (m_1, m_2, m_3, ..., m_n) with n and m_i representing the number of process steps and parallel PMs at Step i, respectively. LLs are classified as Steps 0 and n+1, storing raw and processed wafers. The robot arm with two dirty fingers is termed the dirty arm (A_1), while the arm with two clean fingers is termed the clean arm (A_2). A_1 removes two raw wafers from an LL and transfers them to a PM at Step 1 for processing. After processing, these wafers are marked as clean. A_2 transfers them through Steps 2 to n for further processing and returns them to an LL once all steps are completed.

Scheduling a cluster tool entails determining the robot task sequence to optimize productivity while meeting RTCs. The robot tasks include unloading, loading, and moving; U_i and L_i represent the tasks of unloading and loading at a PM of Step i, $i \in \boldsymbol{N}_n = \{0, 1, \ldots, n\}$, respectively. Given that the time required for a robot task is quite short in practice, the time to perform U_i, $i \in \boldsymbol{N}_n \backslash \{0\}$, and L_i, $i \in \boldsymbol{N}_n$, is the same and is represented by γ. Note that before the robot unloads two raw wafers from an LL, the wafers must first be aligned, making the time to perform U_0 relatively longer. The time taken for U_0 is denoted as γ_0. The rotation task from a PM at Step i (or an LL at Step 0) to a PM at Step j (or an LL at Step 0) is denoted by $R_{i,j}$, i, j, $\in \boldsymbol{N}_n$, and the time taken for $R_{i,j}$ is denoted by β. Further, a swap operation of the robot includes unloading, rotating, and loading tasks. Assume a robot arm is in front of a PM, if it performs a swap operation, the other arm should rotate to the front afterward.

Under the idle state, all four spaces in a PM are empty. After two wafers are loaded into two spaces, the PM's internal chamber rotates to position the other two empty spaces facing the door, allowing the robot to load two more wafers. Likewise, a rotation is required for the robot to remove the four processed wafers from the PM consecutively. Symbol υ represents the time required for a rotation operation of a PM. The wafer processing time in a PM at Step i is denoted by α_i, $i \in \boldsymbol{N}_n \backslash \{0\}$. Assume that there are four wafers in a PM at Step i. Let $W_{i,12}$ and $W_{i,34}$ represent the four wafers processed concurrently in a PM at Step i, with $W_{i,12}$ being the two wafers loaded into the PM before $W_{i,34}$. Once $W_{i,12}$ are loaded in the PM at Step i, the inner chamber turns to permit $W_{i,34}$ to be inserted. Subsequently, after $W_{i,34}$ is loaded, the chamber should rotate again to reposition the two spaces containing $W_{i,12}$ to the door position such that the PM can begin processing the wafers. After the processing activity is completed, $W_{i,12}$ can be removed from the PM earlier than $W_{i,34}$. This sequence ensures that wafers loaded into the PM first are also removed first.

In a steady state, four wafers are processed simultaneously in a PM at Step i. When the robot arrives to unload $W_{i,12}$, the PM may not finish processing, requiring the robot to wait and such a robot waiting task is denoted by T_{i1}. Once $W_{i,12}$ is taken out of the PM, the inner chamber should rotate so that the spaces containing $W_{i,34}$ move to the door position of the PM. Additionally, when the robot returns to the PM for unloading $W_{i,34}$, the chamber rotation might still be in progress. If the robot arrives before this rotation is complete, it should wait again and such a waiting is denoted by T_{i2}. Next, when the robot loads $W_{i,12}$ into a PM at Step i, the PM is empty, allowing immediate loading. After $W_{i,12}$ is loaded, the PM rotates to position the other two empty spaces at the door. When the robot returns to the PM to load $W_{i,34}$ into the PM, the internal chamber of the PM may not finish its rotation, leading to the robot waiting at the PM. This waiting activity is denoted by T_{i3}. The time for T_{i1}, T_{i2}, and T_{i3} is represented by ω_{i1}, ω_{i2}, and ω_{i3}, respectively.

By RTCs, wafers need to be removed from the PM shortly after processing. Let δ_i represent the allowable time at Step i, $i \in \boldsymbol{N}_n \backslash \{0\}$, and d_i the delay for a wafer at Step i, $i \in \boldsymbol{N}_n \backslash \{0\}$, *i.e.*, the interval from the end of processing to the time point of wafer unloading. $\tau_{i,12}$ and $\tau_{i,34}$ represent the residence time of $W_{i,12}$ and $W_{i,34}$, respectively. For an SF^3-CT, the following constraints must be met.

$$\alpha_i \leq \tau_{i,12} \leq \alpha_i + \delta_i,\ i \in \boldsymbol{N}_n \backslash \{0\} \tag{2.1}$$

$$\alpha_i \leq \tau_{i,34} \leq \alpha_i + \delta_i,\ i \in \boldsymbol{N}_n \backslash \{0\} \tag{2.2}$$

B. Robot Task Sequences

For an SF^3-CT with $n \geq 2$, maximizing productivity requires fully loading wafers into each PM under the steady state. To achieve this, a periodic robot task sequence must be determined to ensure both maximum productivity and operational feasibility. [7] and [12] demonstrate that backward and swap strategies are the optimal scheduling strategies for single-arm and dual-arm cluster tools, respectively. For an SF^3-CT, the arm with dirty fingers can be applied to transfer raw wafers only, leading to the swap strategy is not applicable. Thus, This study introduces a hybrid strategy by combining backward and swap strategies and examines the backward strategy for SF^3-CTs, noting its ability to use clean and dirty fingers correctly. This will be explained later. Besides, the performance of SF^3-CTs is analyzed through two robot task sequences.

The first robot task sequence is a backward strategy known as *backward sequence* (BS). For BS, the clean robot arm A_2 first executes task sequence $\kappa_1 = \langle U_n \rightarrow R_{n,n+1} \rightarrow L_{n+1} \rightarrow R_{n+1,n} \rightarrow U_n \rightarrow R_{n,n+1} \rightarrow L_{n+1} \rightarrow R_{n+1,n-1} \rightarrow \ldots \rightarrow R_{3,1} \rightarrow U_1 \rightarrow R_{1,2} \rightarrow L_2 \rightarrow R_{2,1} \rightarrow U_1 \rightarrow R_{1,2} \rightarrow L_2\rangle$ to transfer clean wafers. Then, the dirty robot arm A_1 executes robot task sequence $\kappa_2 = \langle R_{2,0} \rightarrow U_0 \rightarrow R_{0,1} \rightarrow L_1 \rightarrow R_{1,0} \rightarrow U_0 \rightarrow R_{0,1} \rightarrow L_1 \rightarrow R_{1,n}\rangle$ to load dirty wafers into the system. Note that by $R_{1,n}$ in κ_2, a state is reached where A_2 is positioned at a PM of Step n. Therefore, a cycle is finished by sequentially performing κ_1 and κ_2. Notice that, by BS, after performing κ_1, a PM at Step 1 is empty.

The second robot task sequence is a hybrid strategy known as *hybrid task sequence* (HTS). For HTS, A_2 first executes a robot task sequence $\kappa_3 = \langle U_n \rightarrow R_{n,n+1} \rightarrow L_{n+1} \rightarrow R_{n+1,n} \rightarrow U_n \rightarrow R_{n,n+1} \rightarrow L_{n+1} \rightarrow R_{n+1,n-1} \rightarrow \ldots \rightarrow R_{4,2} \rightarrow U_2 \rightarrow R_{2,3} \rightarrow L_3 \rightarrow R_{3,2} \rightarrow U_2 \rightarrow R_{2,3} \rightarrow L_3\rangle$ to transfer clean wafers. Then, A_1 and A_2 collaboratively perform robot task sequence $\kappa_4 = \langle R_{3,0} \rightarrow U_0 \rightarrow R_{0,1} \rightarrow$ a swap operation $\rightarrow R_{1,2} \rightarrow L_2 \rightarrow R_{2,0} \rightarrow U_0 \rightarrow R_{0,1} \rightarrow$ a swap operation $\rightarrow R_{1,2} \rightarrow L_2 \rightarrow R_{2,n}\rangle$ to load dirty wafers into the system. Note that by κ_4, a state is reached where A_2 is positioned at a PM of Step n. Therefore, a production cycle is finished by sequentially performing κ_3 and κ_4. Unlike BS, after performing κ_3, a PM at Step 2 is emptied, but no PM at Step 1 is empty.

These two robot task sequences ensure the correct use of clean and dirty fingers. In κ_1–κ_4, robot waiting time is not considered. Although the task sequence for each of κ_1 to κ_4 is defined, it does not provide a schedule because the start and end times of the activities are not specified. Therefore, to obtain an optimal schedule for either sequence, we need to increase ω_{i1} intentionally. The key to schedule an SF^3-CT is to determine the robot waiting time to meet RTCs and maximize productivity. This will be discussed in the next section.

III. Scheduling Analysis

A. Time Properties under BS

Let θ_i represent the time required to process four wafers at Step i, $i \in \boldsymbol{N}_n\backslash\{0\}$. Under the steady state, to complete four wafers in a PM at Step n, by BS and considering the robot waiting activities, the task sequence is $\kappa_5 = \langle U_n \rightarrow R_{n,n+1} \rightarrow L_{n+1} \rightarrow R_{n+1,n} \rightarrow T_{n2} \rightarrow U_n \rightarrow R_{n,n+1} \rightarrow L_{n+1} \rightarrow R_{n+1,n-1} \rightarrow T_{(n-1)1} \rightarrow U_{n-1} \rightarrow R_{n-1,n} \rightarrow L_n \rightarrow R_{n,n-1} \rightarrow T_{(n-1)2} \rightarrow U_{n-1} \rightarrow R_{n-1,n} \rightarrow T_{n3} \rightarrow L_n \rightarrow$ the internal chamber rotates $\rightarrow$ wafers are processed $\rightarrow$ four wafers remain after processing$\rangle$. With m_n parallel PMs serving for Step n, θ_n can be calculated according to κ_5 as follows.

$$\theta_n = (\alpha_n + d_n + 8\gamma + 7\beta + \upsilon + \omega_{n2} + \omega_{n3} + \omega_{(n-1)1} + \omega_{(n-1)2})/m_n \quad (3.1)$$

In addition, to complete four wafers in a PM of Step i, $i\in\{3, ..., n - 1\}$ and $n > 2$, by BS and considering the robot waiting activities, the task sequence is $\kappa_6 = \langle U_i \rightarrow R_{i,i+1} \rightarrow L_{i+1} \rightarrow R_{i+1,i} \rightarrow T_{i2} \rightarrow U_i \rightarrow R_{i,i+1} \rightarrow T_{(i+1)3} \rightarrow L_{i+1} \rightarrow R_{i+1,i-1} \rightarrow T_{(i-1)1} \rightarrow U_{i-1} \rightarrow R_{i-1,i} \rightarrow L_i \rightarrow R_{i,i-1} \rightarrow T_{(i-1)2} \rightarrow U_{i-1} \rightarrow R_{i-1,i} \rightarrow T_{i3} \rightarrow L_i \rightarrow$ the internal chamber rotates $\rightarrow$ wafers are processed $\rightarrow$ four wafers remain after processing$\rangle$. With m_i parallel PMs serving for Step i and κ_6, θ_i can be calculated by (3.2).

$$\theta_i = (\alpha_i + d_i + 8\gamma+ 7\beta + \upsilon + \omega_{(i+1)3} + \omega_{i2} + \omega_{i3} + \omega_{(i-1)1} + \omega_{(i-1)2})/m_i,\ i \in\{3, ..., n - 1\} \text{ and } n > 2 \quad (3.2)$$

In the same way, to complete four wafers in a PM of Step 1, by BS, with the robot waiting activities being considered, task sequence $\kappa_7 = \langle U_1 \rightarrow R_{1,2} \rightarrow L_2 \rightarrow R_{2,1} \rightarrow T_{12} \rightarrow U_1 \rightarrow R_{1,2} \rightarrow T_{23} \rightarrow L_2 \rightarrow R_{2,0} \rightarrow U_0 \rightarrow R_{0,1} \rightarrow L_1 \rightarrow R_{1,0} \rightarrow U_0 \rightarrow R_{0,1} \rightarrow T_{13} \rightarrow L_1 \rightarrow$ the internal chamber rotates $\rightarrow$ wafers are processed $\rightarrow$ four wafers remain after processing$\rangle$ should be performed. With m_1 parallel PMs serving for Step 1 and κ_7, θ_1 can be calculated by (3.3).

$$\theta_1 = (\alpha_1 + d_1 + 2\gamma_0 + 6\gamma + 7\beta + \upsilon + \omega_{23} + \omega_{12} + \omega_{13})/m_1 \quad (3.3)$$

By observing κ_5, κ_6, and κ_7, it can be concluded that T_{i2}, $i \in \boldsymbol{N}_n\backslash\{0\}$, is caused by the rotation of the internal chamber of a PM for Step i. Note that after the robot completes its first unloading task at the PM in κ_5, κ_6, and κ_7, the internal chamber of the PM begins to rotate. Meanwhile, the robot executes task sequence $\langle R_{i,i+1} \rightarrow L_{i+1} \rightarrow R_{i+1,i}\rangle$. T_{i2} should be executed thereafter, and its duration is determined by the longer of the two: the time required for the rotation operation of the internal chamber of a PM for Step i or the task sequence $\langle R_{i,i+1} \rightarrow L_{i+1} \rightarrow R_{i+1,i}\rangle$. Therefore, we have the following equation.

$$\omega_{i2} = max(\upsilon - (2\beta + \gamma), 0),\ i \in \boldsymbol{N}_n\backslash\{0\} \quad (3.4)$$

In κ_5, κ_6, and κ_7, it can be concluded that T_{i3}, $i \in \boldsymbol{N}_n\backslash\boldsymbol{N}_1$, is caused by the rotation of the internal chamber of a PM for Step i as well. Notice that after the robot performs a loading task at Step i in κ_5, κ_6, and κ_7 for the first time, the internal chamber of the PM begins to rotate. Meanwhile, the robot executes task sequence $\langle R_{i,i-1} \rightarrow T_{(i-1)2} \rightarrow U_{i-1} \rightarrow R_{i-1,i}\rangle$. Thereafter, T_{i3} should be executed and its duration is determined by the longer of the two: the time required for the rotation operation of the internal chamber of a PM for Step i or the task sequence $\langle R_{i,i-1} \rightarrow T_{(i-1)2} \rightarrow U_{i-1} \rightarrow R_{i-1,i}\rangle$. Therefore, we have the following equation.

$$\omega_{i3} = max(\upsilon - (\omega_{(i-1)2} + 2\beta + \gamma), 0),\ i \in \boldsymbol{N}_n\backslash\boldsymbol{N}_1 \quad (3.5)$$

In the same way, ω_{13} is determined by the longer time taken for a rotation operation of a PM for Step 1 and the task sequence $\langle R_{1,0} \rightarrow U_0 \rightarrow R_{0,1}\rangle$, and hence we get the following equation.

$$\omega_{13} = max(\upsilon - (2\beta + \gamma_0), 0) \quad (3.6)$$

Following κ_5 and κ_6, the duration of the task sequence $\langle R_{i,i-1} \rightarrow T_{(i-1)2} \rightarrow U_{i-1} \rightarrow R_{i-1,i} \rightarrow T_{i3} \rightarrow L_i \rightarrow$ the internal chamber rotates $\rightarrow$ wafers are processed $\rightarrow$ four wafers remain after processing⟩ and ⟨the internal chamber rotates → wafers are processed → four wafers remain after processing → U_i → $R_{i,i+1}$ → L_{i+1} → $R_{i+1,i}$ → T_{i2}⟩ is $\tau_{i,12}$ and $\tau_{i,34}$, $i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_1$, respectively. Therefore, we have the following equations.

$$\tau_{i,12} = \alpha_i + d_i + \upsilon + 2\gamma + 2\beta + \omega_{(i-1)2} + \omega_{i3},\ i \in \ \boldsymbol{N}_n \backslash \boldsymbol{N}_1 \quad (3.7)$$

$$\tau_{i,34} = \alpha_i + d_i + \upsilon + 2\gamma + 2\beta + \omega_{i2},\ i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_1 \quad (3.8)$$

Following κ_7, the duration of the task sequence ⟨$R_{1,0}$ → U_0 → $R_{0,1}$ → T_{13} → L_1 → the internal chamber rotates → wafers are processed → four wafers remain after processing⟩ and ⟨the internal chamber rotates → wafers are processed → four wafers remain after processing → U_1 → $R_{1,2}$ → L_2 → $R_{2,1}$ → T_{12}⟩ should be $\tau_{1,12}$ and $\tau_{1,34}$, respectively. Thus, $\tau_{1,12}$ and $\tau_{1,34}$ can be calculated as follows.

$$\tau_{1,12} = \alpha_1 + d_1 + \upsilon + \gamma + \gamma_0 + 2\beta + \omega_{13} \quad (3.9)$$

$$\tau_{1,34} = \alpha_1 + d_1 + \upsilon + 2\gamma + 2\beta + \omega_{12} \quad (3.10)$$

By BS, considering the robot waiting activities, κ_1 and κ_2 can be rewritten as ⟨U_n → $R_{n,n+1}$ → L_{n+1} → $R_{n+1,n}$ → T_{n2} → U_n → $R_{n,n+1}$ → L_{n+1} → $R_{n+1,n-1}$ → … → $R_{3,1}$ → T_{11} → U_1 → $R_{1,2}$ → L_2 → $R_{2,1}$ → T_{12} → U_1 → $R_{1,2}$ → T_{23} → L_2⟩ and ⟨$R_{2,0}$ → U_0 → $R_{0,1}$ → L_1 → $R_{1,0}$ → U_0 → $R_{0,1}$ → T_{13} → L_1 → $R_{1,n}$ → T_{n1}⟩, respectively. Under BS, the robot cycle time ψ_1, resulting from the combination of κ_1 and κ_2, is as follows.

$$\psi_1 = 4(n+1)\beta + 2(2n+1)\gamma + 2\gamma_0 + \sum_1^n \omega_{i1} + \sum_1^n \omega_{i2} + \sum_1^n \omega_{i3} \quad (3.11)$$

B. Time Properties under HTS

By HTS, considering the robot waiting activities, κ_5 or κ_6 is executed to complete four wafers by a PM for Step i, $i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_2$. Hence, with m_i parallel PMs for Step i, based on κ_5 and κ_6, θ_i, $i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_2$, can be calculated by using (3.1) and (3.2).

For Step 2, where $n > 2$, κ_8 = ⟨U_2 → $R_{2,3}$ → L_3 → $R_{3,2}$ → T_{22} → U_2 → $R_{2,3}$ → T_{33} → L_3 → $R_{3,0}$ → U_0 → $R_{0,1}$ → T_{11} → U_1 → the robot rotates → L_1 → $R_{1,2}$ → L_2 → $R_{2,0}$ → U_0 → $R_{0,1}$ → T_{12} → U_1 → the robot rotates → L_1 → $R_{1,2}$ → T_{23} → L_2 → the internal chamber rotates → wafers are processed → four wafers remain after processing⟩ should be executed to complete four wafers at a PM. With m_2 parallel PMs for Step 2, θ_2 can be calculated as follows based on κ_8.

$$\theta_2 = (\alpha_2 + d_2 + 2\gamma_0 + 10\gamma + 11\beta + \upsilon + \omega_{33} + \omega_{22} + \omega_{23} + \omega_{11} + \omega_{12})/m_2,\ n > 2 \quad (3.12)$$

For Step 2, $n = 2$, the robot activity time to complete four wafers at a PM is the same as that of Step 2 when $n > 2$ in the task sequence. However, the waiting time differs due to the absence of Step 3 (T_{33}). With m_2 parallel PMs serving Step 2, θ_2 can be calculated as follows.

$$\theta_2 = (\alpha_2 + d_2 + 2\gamma_0 + 10\gamma + 11\beta + \upsilon + \omega_{22} + \omega_{23} + \omega_{11} + \omega_{12})/m_2,\ n = 2 \quad (3.13)$$

For Step 1, κ_9 = ⟨U_1 → the robot rotates → L_1 → $R_{1,2}$ → L_2 → $R_{2,0}$ → U_0 → $R_{0,1}$ → T_{12} → U_1 → the robot rotates → L_1 → the internal chamber rotates → wafers are processed → four wafers remain after processing⟩ should be executed to complete four wafers at a PM. With m_1 parallel PMs for Step 1, θ_1 can be calculated as follows based on κ_9.

$$\theta_1 = (\alpha_1 + d_1 + \gamma_0 + 5\gamma + 5\beta + \upsilon + \omega_{12})/m_1 \quad (3.14)$$

The robot waiting activity T_{i2}, $i \in \boldsymbol{N}_n \backslash \{0\}$, is caused by the rotation of the PM's internal chamber for Step i. Also, ω_{i2}, $i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_1$, is determined by the longer duration between the PM's rotation operation and ⟨$R_{i,i+1}$ → L_{i+1} → $R_{i+1,i}$⟩. Therefore, ω_{i2}, $i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_1$, can be calculated by (3.4).

In κ_9, after the robot completes its first loading task at Step 1 and finishes a rotation, the internal chamber of the PM begins to rotate. Simultaneously, the robot executes the task sequence ⟨$R_{1,2}$ → L_2 → $R_{2,0}$ → U_0 → $R_{0,1}$⟩. Therefore, ω_{12} is determined by the longer time of the two including the time taken for the PM's rotation operation and task sequence ⟨$R_{1,2}$ → L_2 → $R_{2,0}$ → U_0 → $R_{0,1}$⟩. Then, ω_{12} is obtained as follows.

$$\omega_{12} = max(\upsilon - (\gamma + \gamma_0 + 3\beta), 0) \quad (3.15)$$

T_{i3}, $i \in \boldsymbol{N}_n \backslash \{0\}$, is also caused by the rotation of the PM's internal chamber for Step i. Then, ω_{i3}, $i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_2$, is determined by the longer duration between the PM's rotation operation for Step i and task sequence ⟨$R_{i,i-1}$ → $T_{(i-1)2}$ → U_{i-1} → $R_{i-1,i}$⟩. Therefore, ω_{i3}, $i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_2$, can be calculated by (3.5).

In κ_8, after the robot completes its first loading task at Step 2, the PM's internal chamber begins to rotate. Simultaneously, the robot executes the task sequence ⟨$R_{2,0}$ → U_0 → $R_{0,1}$ → T_{12} → U_1 → the robot rotates → L_1 → $R_{1,2}$⟩. Hence, ω_{23} is determined by the longer one between the time taken for the PM's rotation operation and task sequence ⟨$R_{2,0}$ → U_0 → $R_{0,1}$ → T_{12} → U_1 → the robot rotates → L_1 → $R_{1,2}$⟩. Therefore, (3.16) is given to calculate ω_{23}.

$$\omega_{23} = max(\upsilon - (2\gamma + \gamma_0 + 4\beta + \omega_{12}), 0) \quad (3.16)$$

Note that T_{13} represents the robot waiting activity at a PM for Step 1 before loading the wafers represented by $W_{1,34}$ into the PM. In κ_9, because the robot executes a swap operation to load these wafers into the PM, T_{13} is unnecessary.

Furthermore, similar to BS, for Step i, $i \in \boldsymbol{N}_n \backslash \boldsymbol{N}_2$, $\tau_{i,12}$ and $\tau_{i,34}$ can be calculated by (3.7) and (3.8), respectively. For Step 2, based on observing κ_8, the time required for task sequences ⟨$R_{2,0}$ → U_0 → $R_{0,1}$ → T_{12} → U_1 → the robot rotates → L_1 → $R_{1,2}$ → T_{23} → L_2 → the internal chamber rotates → wafers are processed → four wafers remain after processing⟩ and ⟨the internal chamber rotates → wafers are processed → four wafers remain after processing → U_2 → $R_{2,3}$ → L_3 → $R_{3,2}$ → T_{22}⟩ are supposed to be $\tau_{2,12}$ and $\tau_{2,34}$, respectively. Afterward, (3.17) is given to obtain $\tau_{2,12}$, while $\tau_{2,34}$ can be determined by (3.8).

$$\tau_{2,12} = \alpha_2 + d_2 + \upsilon + \gamma_0 + 3\gamma + 4\beta + \omega_{12} + \omega_{23} \quad (3.17)$$

For Step 1, based on κ_9, the time required for task sequences ⟨$R_{1,2}$ → L_2 → $R_{2,0}$ → U_0 → $R_{0,1}$ → T_{12} → U_1 → the robot rotates → L_1 → the internal chamber rotates → wafers are processed → four wafers remain after processing⟩ and ⟨the internal chamber rotates → wafers are processed → four wafers remain after processing → U_1 → the robot rotates → L_1 → $R_{1,2}$ → L_2 → $R_{2,0}$ → U_0 → $R_{0,1}$ → T_{12}⟩ are supposed to be $\tau_{1,12}$ and $\tau_{1,34}$, respectively. Therefore, we derive the following equation.

$$\tau_{1,12} = \tau_{1,34} = \alpha_1 + d_1 + \upsilon + 3\gamma + \gamma_0 + 4\beta + \omega_{12} \quad (3.18)$$

By HTS, considering the robot waiting activities, κ_3 and κ_4 can be rewritten as $\langle U_n \rightarrow R_{n,n+1} \rightarrow L_{n+1} \rightarrow R_{n+1,n} \rightarrow T_{n2} \rightarrow U_n \rightarrow R_{n,n+1} \rightarrow L_{n+1} \rightarrow R_{n+1,n-1} \rightarrow \ldots \rightarrow R_{4,2} \rightarrow T_{21} \rightarrow U_2 \rightarrow R_{2,3} \rightarrow L_3 \rightarrow R_{3,2} \rightarrow T_{22} \rightarrow U_2 \rightarrow R_{2,3} \rightarrow T_{33} \rightarrow L_3\rangle$ and $\langle R_{3,0} \rightarrow U_0 \rightarrow R_{0,1} \rightarrow T_{11} \rightarrow$ a swap operation at Step 1 $\rightarrow R_{1,2} \rightarrow L_2 \rightarrow R_{2,0} \rightarrow U_0 \rightarrow R_{0,1} \rightarrow T_{12} \rightarrow$ a swap operation at Step 1 $\rightarrow R_{1,2} \rightarrow T_{23} \rightarrow L_2 \rightarrow R_{2,n}\rangle$, respectively. Next, under HTS, the robot cycle time ψ_2 resulting from the combination of κ_3 and κ_4 can be obtained as

$$\psi_2 = 4(n + 1)\beta + 2(2n + 1)\,\gamma + 2\gamma_0 + \textstyle\sum_1^n \omega_{i1} + \sum_1^n \omega_{i2} + \sum_1^n \omega_{i3} \qquad (3.19)$$

C. Linear Programs for Solution

By either BS or HTS, if an SF^3-CT operates under a steady state, each step in a robot task cycle processes four wafers, which are then returned to the LLs upon completion. Additionally, the time required to complete four wafers at each step equals the robot cycle time. Based on this, we derive Proposition 3.1.

Proposition 3.1: For an SF^3-CT operated by BS or HTS under the steady state, the robot cycle time equals the time to complete four wafers per step, as stated in (3.20).

$$\psi_k = \theta_i,\ k = \{1, 2\},\ i \in \boldsymbol{N}_n\backslash\{0\} \qquad (3.20)$$

(2.1) and (2.2) are supposed to be satisfied under either of the two strategies to fulfill RTCs. Given that the robot task sequences are predetermined, minimizing the robot waiting time within these sequences is crucial. Note that ω_{i2} and ω_{i3} are deterministic. The key is to determine ω_{i1} after a strategy is chosen. By Proposition 3.1, we develop two linear programs for two strategies to find feasible schedules that minimize the cycle time.

Linear Program 1 (**LP1**)**:** For an SF^3-CT operated by BS under the steady state, a linear program is established to get a schedule as stated below.

Minimize ψ_1

Subject to: (2.1), (2.2), (3.1)–(3.11), (3.20).

In LP1, (2.1) and (2.2) ensure that the RCTs are satisfied. (3.1)–(3.3) calculate θ_i, $i \in \boldsymbol{N}_n\backslash\{0\}$. Then, (3.4)–(3.6) calculate ω_{i2} and ω_{i3} at each step. (3.7)–(3.10) determine $\tau_{i,12}$ and $\tau_{i,34}$. (3.11) gets the robot cycle time under BS. (3.20) from Proposition 3.1 helps get ω_{i1} and ψ_1.

Linear Program 2 (**LP2**)**:** For an SF^3-CT operated by HTS under the steady state, a linear program is established to get a schedule as stated below.

Minimize ψ_2

Subject to: (2.1), (2.2), (3.1), (3.12)–(3.20)

(3.2) where $i \in \{2, 3, ..., n - 1\}$ and $n > 3$,

(3.4) and (3.8), where $i \in \boldsymbol{N}_n\backslash\boldsymbol{N}_1$, and

(3.5) and (3.7), where $i \in \boldsymbol{N}_n\backslash\boldsymbol{N}_2$.

In LP2, (2.1) and (2.2) ensure that the RCTs are satisfied. (3.1), (3.2), (3.12)–(3.14) calculate θ_i, $i \in \boldsymbol{N}_n\backslash\{0\}$. Then, (3.4), (3.5), (3.15), and (3.16) calculate ω_{i2} and ω_{i3} at each step. (3.7), (3.8), (3.17), and (3.18) determine $\tau_{i,12}$ and $\tau_{i,34}$. (3.19) gives the robot cycle time under HTS. (3.20) from Proposition 3.1 helps get ω_{i1} and ψ_2.

It is important to highlight that applying different strategies with LPs 1 and 2 may yield different results due to varying parameter combinations. Therefore, given a set of parameters by LPs 1 and 2, the better strategy can be chosen.

IV. Experiments

Experiments with different parameters are conducted to evaluate the two scheduling strategies by running LPs 1 and 2 under various parameters. The results are presented as follows. The measurement of time is seconds and omitted in subsequent cases.

***Example* 1**: The WFP in this example is (1, 1, 1). Steps 1 to 3 are used for PM_1 to PM_3, respectively. Here are three scenarios.

Case 1: The processing time α_1 for Step 1 is randomly selected from the range [90.0, 170.0]. Also, α_2 = 110.0, α_3 = 115.0, δ_1 = 30.0, δ_2 = 30.0, and δ_3 = 35.0. The time duration of robot task is γ = 3.0, γ_0 = 4.0, and β = 2.0. The PM rotation time is υ = 3.0. In this case, α_1 varies while all other parameters remain constant. The results are displayed in Fig. 2.

Case 2: The processing time is α_1 = 135.0, α_2 is randomly selected from the range [90.0, 150.0]. Also, α_3 = 122.0, δ_1 = 30.0, δ_2 = 30.0, and δ_3 = 20.0. The time duration of robot task is γ = 2.0, γ_0 = 3.0, and β = 2.0. The PM rotation time is υ = 8.0. In this case, α_2 varies while all other parameters remain constant. We have the results displayed in Fig. 3.

Case 3: The processing time is α_1 = 121.0, α_2 = 95.0, and α_3 is randomly selected from the range [90.0, 150.0]. And δ_1 = 25.0, δ_2 = 25.0, δ_3 = 20.0. The time duration of robot task is γ = 2.0, γ_0 = 3.0, β = 2.0. The PM rotation time is υ = 5.0. In this case, α_3 varies while all other parameters remain constant. We have the results displayed in Fig. 4.

We evaluate LPs 1 and 2 and determine the ranges of parameter variation where feasible schedules can be identified, as illustrated in Figs. 2–4. From these figures, BS can identify schedules within the larger parameter variation ranges. When the processing time of Step 1 is relatively long, HTS has a shorter cycle time, but the range of parameter variation resulting in feasible schedules is narrower compared to BS.

***Example* 2**: The WFP in this example is (2, 1, 1). PM_1 and PM_2 are used for Step 1, and PM_3 and PM_4 are used for Steps 2 and 3, respectively. Here are three scenarios.

Case 1: The processing time is α_1 = 275.0, α_2 = 100.0, α_3 = 109.0. δ_1 is randomly selected from [10.0, 30.0]. Also, δ_2 = 32.0, and δ_3 = 29.0. The time duration of robot task is γ = 2.0, γ_0 = 4.0, and β = 2.0. The PM rotation time is υ = 10.0. Based on LPs 1 and 2, when δ_1 = 31.0 and 27.0, BS and HTS find feasible schedules with the system cycle time ψ_1 = 162.5, ψ_2 = 164.0, respectively. With an increase in δ_1, HTS initially finds a feasible schedule, while BS demonstrates a shorter cycle time.

Case 2: Compared to Case 1, υ is decreased from 10 to 4, δ_1 is randomly selected from [10.0, 30.0], while the remaining parameters remain the same. Based on LPs 1 and 2, when $\upsilon = 4.0$, and δ_1 is 14.0 and 37.0, BS and HTS identify schedulable system cycle time $\psi_1 = 156.5$ and $\psi_2 = 154.0$, respectively. When the PM rotation time is shorter, it is shown that BS performs better with strict RTCs, whereas HTS attains a shorter cycle time but needs more lenient RTCs.

Case 3: The wafer processing time is $\alpha_1 = 315.0$, $\alpha_2 = 120.0$, $\alpha_3 = 128.0$, $\delta_1 = 36.0$, $\delta_2 = 32.0$, and $\delta_3 = 34.0$. The robot task time is $\gamma = 2.0$, $\gamma_0 = 4.0$, and $\beta = 2.0$. And let υ range from 2.0 to 16.0. Based on LPs 1 and 2, Fig. 5 shows that HTS performs better when the PM internal chamber rotation time is shorter. As a PM's internal chamber rotation time increases, BS exhibits a wider scheduling range within the fluctuation range of υ.

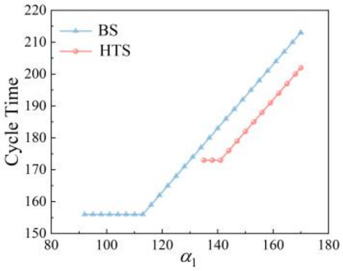

Figure 2. Cycle time varies with α_1

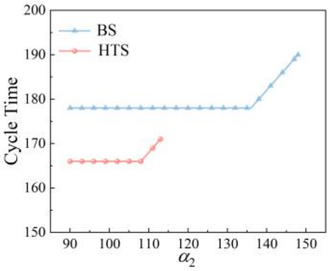

Figure 3. Cycle time varies with α_2

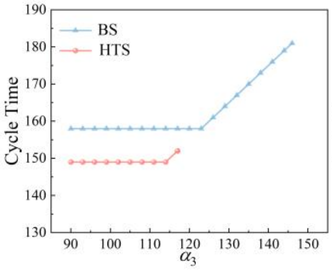

Figure 4. Cycle time varies with α_3

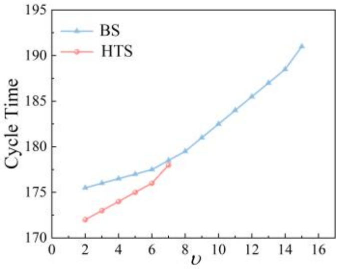

Figure 5. Cycle time varies with υ

V. CONCLUSIONS

This study investigates the scheduling issue of SF^3-CTs, a type of multi-finger-robotic cluster tools with multi-space process modules used in modern fabs. Considering the configuration properties, the wafer staying time in PMs is influenced by both the robot task sequences and the rotation operations of the PMs. The additional time required within PMs for handling multiple spaces makes meeting the residency time constraints more difficult. We present two scheduling strategies to address this issue and develop two linear programs to identify feasible schedules. Experiments are conducted to evaluate BS and HTS. The outcomes show that one scheduling strategy cannot be superior for all cases. Thus, given a set of parameters, both strategies should be applied, and a better schedule should be selected.

Next, we will concentrate on the scheduling issue of SF^3-CTs with chamber cleaning requirements and simultaneously processing multiple types of wafers.

REFERENCES

[1] T. K. McNab, “Cluster tools, part 1: Emerging processes,” Semicond. Int., vol. 13, no. 9, pp. 58–63, Aug. 1990.

[2] P. Singer, “The driving forces in cluster tool development,” Semicond. Int., vol. 18, no. 8, pp. 113–118, Jul. 1995.

[3] 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.

[4] S. Rostami, B. Hamidzadeh, and D. Camporese, “An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints,” IEEE Trans. Robot. Autom., vol. 17, no. 5, pp. 609–618, Oct. 2001.

[5] M.-J. Lopez and S.-C. Wood, “Systems of multiple cluster tools: Configuration, reliability, and performance,” IEEE Trans. Semicond. Manuf., vol. 16, no. 2, pp. 170–178, May 2003.

[6] T.-E. Lee, H.-Y. Lee, and Y.-H. Shin, “Workload balancing and scheduling of a single-armed cluster tool,” in Proc. 5th APIEMS Conf., Gold Coast, QLD, Australia, 2004, pp. 1–15.

[7] 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. Semicond. Manuf., vol. 10, pp. 418–424, 1997.

[8] J. Wang, H. Hu, C. Pan, Y. Zhou, and L. Li, “Scheduling dual-arm cluster tools with multiple wafer types and residency time constraints,” IEEE/CAA J. Automatica Sinica, vol. 7, no. 3, pp. 776–789, May 2020.

[9] J. Yi, S. Ding, D. Song, and M. T. Zhang, “Steady-state throughput and scheduling analysis of multi-cluster tools: A decomposition approach,” IEEE Trans. Autom. Sci. Eng., vol. 5, no. 2, pp. 321–336, Apr. 2008.

[10] S. Ding, J. Yi, and M. T. Zhang, “Multicluster tools scheduling: An integrated event graph and network model approach,” IEEE Trans. Semicond. Manuf., vol. 19, no. 3, pp. 339–351, Aug. 2006.

[11] T. L. Perkinson, P. K. McLarty, R. S. Gyurcsik, and R. K. Cavin, “Single-wafer cluster tool performance: An analysis of throughput,” IEEE Trans. Semicond. Manuf., vol. 7, no. 3, pp. 369–373, Aug. 1994.

[12] W. K. V. Chan, J. Yi, and S. Ding, “Optimal scheduling of multicluster tools with constant robot moving times, part I: Two-cluster analysis,” IEEE Trans. Autom. Sci. Eng., vol. 8, no. 1, pp. 5–16, Jan. 2011.

[13] J.-H. Kim, T.-E. Lee, H.-Y. Lee, and D.-B. Park, “Scheduling analysis of time-constrained dual-armed cluster tools,” IEEE Trans. Semicond. Manuf., vol. 16, no. 3, pp. 521–534, Aug. 2003.

[14] J.-H. Lee, H.-J. Kim, and T.-E. Lee, “Scheduling lot switching operations for cluster tools,” IEEE Trans. Semicond. Manuf., vol. 26, no. 4, pp. 592–601, Nov. 2013.

[15] Y. Qiao, M. Zhou, N. Wu, and Q. 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.

[16] Y. Lim, T.-S. Yu, and T.-E. Lee, “A new class of sequences without interferences for cluster tools with tight wafer delay constraints,” IEEE Trans. Autom. Sci. Eng., vol. 16, no. 1, pp. 392–405, Jan. 2019.

[17] S.-G. Ko, T.-S. Yu, and T.-E. Lee, “Wafer delay analysis and workload balancing of parallel chambers for dual-armed cluster tools with multiple wafer types,” IEEE Trans. Autom. Sci. Eng., vol. 18, no. 3, pp. 1516–1526, Jul. 2021.

[18] Y. Lim, T.-S. Yu, and T.-E. Lee, “Adaptive scheduling of cluster tools with wafer delay constraints and process time variation,” IEEE Trans. Autom. Sci. Eng., vol. 17, no. 1, pp. 375–388, Jan. 2020.

[19] Q. Zhu, G. Wang, Y. Hou, N. Wu, and Y. Qiao, “Optimally Scheduling Dual-Arm Multi-Cluster Tools to Process Two Wafer Types,” IEEE Trans. Robot. Autom., vol. 7, no. 3, pp. 5920–5927, Jul. 2022.

[20] 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.

[21] Y. Qiao, N. Q. Wu, and M. C. Zhou, “Schedulability and scheduling analysis of dual-arm cluster tools with wafer revisiting and residency time constraints based on a novel schedule,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 45, no. 3, pp. 472–484, Mar. 2015.

[22] Y. Qiao, M. Zhou, N. Wu, Z. Li, and Q. Zhu, “Closing-Down Optimization for Single-Arm Cluster Tools Subject to Wafer Residency Time Constraints,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 51, no. 11, pp. 6792–6807, Nov. 2021