Journal of Industrial and Management Optimization

JIMO, 22(4): 1956–1980.
DOI: 10.3934/jimo.2026072
Received: 21 November 2025
Revised: 27 February 2026
Accepted: 10 March 2026
Published: 23 March 2026

https://www.aimspress.com/journal/jimo

Research article

An Efficient Mixed Integer Programming Model for Wet Station Scheduling Problems Based on Timed Petri Nets

Luetao Li[1], Naiqi Wu[1], Yan Qiao[1], Siwei Zhang[1,*], Jie Li[2] and Yang Bai[3]

[1] Institute of Systems Engineering and Collaborative Laboratory for Intelligent Science and Systems, Macau University of Science and Technology, Taipa, Macao 999078, Macao, China
[2] IKAS Industries Company, Ltd., Beijing 100176, China
[3] The Information Science Academy, China Electronics Technology Group Corporation, Beijing 100043, China

* **Correspondence:** Email: swzhang@must.edu.mo.

Abstract: In semiconductor wafer production, a cleaning operation is essential for removing surface residues left on wafers after various processing steps. This step takes up approximately 30% of the total fabrication time, making it as important as other major production stages. A wet station is typically composed of a transport single-arm robot and several processing tanks that are arranged in a linear topology configuration. In contrast to cluster tools whose processing modules are arranged in a radial layout, the time taken for the robot to move between tanks in a cleaning tool varies and cannot be considered identical. Under such circumstances, different assignment schemes of tanks to cleaning steps and robot access sequences to the tanks at a step can influence the performance of a wafer cleaning tool, which further complicates the scheduling problem. Our main objective of this paper was to identify the optimal arrangement of allocating tanks to processing steps and the best access route of the robot under various scenarios for maximizing the cleaning efficiency of a wet station. To achieve this goal, a Mixed Integer Programming (MIP) model based on Timed Petri Nets (TPN) was established to determine the optimal cyclic schedule along with the corresponding assignment scheme and cycle time. The practicality of the proposed model was further validated through experimental studies. Our results demonstrated that the proposed model can significantly improve the performance, reduces the average cycle time by 6.66%, and increases throughput by 7.14% compared to traditional scheduling methods. Moreover, the MIP solver obtained optimal schedules within minutes for typical wet station scales, confirming its computational efficiency and practical applicability.

Keywords: wet station; mixed integer programming; scheduling; wafer fabrication; Petri nets
Mathematics Subject Classification: 90-08

1. Introduction

In semiconductor wafer manufacturing, a cleaning process should be performed to wafers that undergo some fabrication steps to effectively remove the surface residues with a constraint of not causing any damage to the wafers, implying that wafer cleaning is a vital step in wafer fabrication. To ensure effective cleaning, a wafer cleaning tool needs to be carefully scheduled and controlled to maximize productivity and minimize cost while ensuring quality. The so-called wet stations are the primarily adopted wafer cleaning tools [1]. Depending on the required level of processing precision, cleaning techniques are typically categorized into single wafer cleaning and batch wafer cleaning. Single wafer cleaning is typically applied when extremely high precision is required, whereas batch wafer cleaning that cleans multiple wafers simultaneously is more efficient in terms of time as well material usage and is suitable for general precision requirements. Owing to the market demand, most wafer cleaning processes adopt a batch method, which our focus of this study. Figure 1 illustrates a wet station used for batch wafer cleaning.

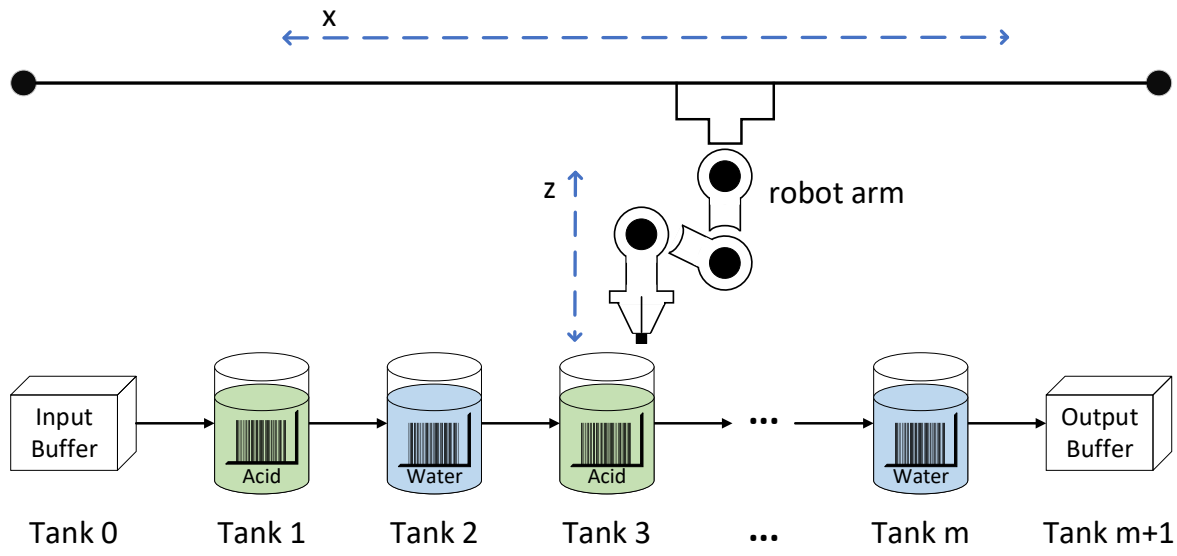

Figure 1. The illustration of a wet station.

As displayed in Figure 1, a wet station (also called automated bench cleaning tool) comprises a number of processing tanks, an input buffer, an output buffer, and a hoist equipped with a single robotic arm (called robot in short). These components are arranged linearly with the input and output buffers located at the two ends. The buffers are used to hold Front Opening Unified Pods (FOUPs), and each FOUP typically contains 25 wafers. The input buffer holds wafers (FOUPs) waiting to be cleaned, while the output buffer stores wafers whose cleaning process has been completed. Wafer cleaning involves a number of sequential processing steps, and each step is carried by a number of dedicated tanks with specified liquid in them as shown in Figure 1. The robot is responsible for transporting

FOUPs between tanks according to a predefined processing sequence. Initially, raw wafers in an FOUP are picked up from the input buffer and transferred by the robot to the appropriate tanks for processing. After all the required cleaning steps are completed, the wafers in an FOUP are delivered to the output buffer. It is significant to note that, for a given processing step, multiple tanks (referred to as parallel tanks) may be equipped to serve the cleaning task.

Many wafer fabrication processes are performed by cluster tools, such as etching, chemical vapor deposition, and physical vapor deposition, where the processing modules (PMs) are arranged radially, and the robot moving time between any two PMs can be treated as an identical constant [2,3]. In contrast, cleaning tools adopt a linear configuration, resulting in varying distances between tanks. Therefore, the robot's movement time between pairs of tanks in a cleaning tool is different, which greatly increases the complexity to the analysis of scheduling such a tool. On the one hand, for a cleaning process that involves multiple steps with some of these steps equipped with parallel tanks, distinct tank assignments to the processing steps can cause cycle time variations under a cyclic schedule due to the differences of robot moving time. This implies that properly assigning parallel tanks to the cleaning steps can enhance cleaning efficiency. On the other hand, when a cleaning step is served by multiple parallel tanks, the order for the robot to visit these tanks in a step can lead to different overall cycle time. Consequently, selecting an appropriate access sequence by taking the distances between the tanks into account becomes crucial for optimizing system performance.

In the context of the rapid development of the semiconductor industry, wafer fabrication facilities increasingly seek efficient scheduling strategies to meet demand within minimal processing time while delivering cost-effective and high-quality products, which has become a key source of competitive advantage. Borodin et al. [4] provide a comprehensive overview of scheduling challenges commonly encountered in semiconductor manufacturing facilities, while Huang et al. [5] survey the opportunities and challenges arising from the integration of machine learning techniques into semiconductor scheduling. As a critical component of modern semiconductor production systems, the wet station needs to be scheduled to operate efficiently under limited resources in order to minimize task completion time. Researchers studying wet-station scheduling have primarily focused on reducing cycle time to improve cleaning efficiency, employing approaches such as mixed-integer programming [6], heuristic algorithms [7], and machine learning [8]. However, in these studies, the impacts of robot access order to parallel chambers and the allocation of cleaning tanks to processing steps on the performance receive limited attention. Our preliminary analysis indicates that both factors may influence the steady-state cycle time of a wet station. With the above discussion in mind, given a cleaning process, we address two issues: (1) Determining the optimal tank assignment to the steps, and (2) refining the order for the robot to visit the parallel tanks at each step for the minimization of tool cycle time. Our goal is to show the impact of these two issues on the performance of such a tool by optimizing the tank allocation to steps and the robot visiting sequence to the parallel tanks at each step. Thus, after extensive literature review and continuous investigation, we combine timed Petri net (TPN) modeling and the mixed integer programming (MIP) formulation to solve the scheduling problem of cleaning tools.

Compared with existing wet station scheduling approaches, this study makes the following key contributions: (1) We develop a flexible cyclic scheduling model for wet stations, which can be readily adapted to wafer cleaning recipes; (2) the proposed model introduces essential decision variables and associated constraints to determine the optimal tank assignment to processing steps and robot visiting sequence to the parallel tanks at each step across different production scenarios; and (3) the model

achieves optimal solutions with short computational time and shows that, by doing so, the performance can be significantly improved over conventional scheduling methods in most cases.

The remaining sections of the paper are organized as follows: In the next section, we provide a detailed review of literature related to the scheduling of cluster tools and wet stations in semiconductor manufacturing. In Section 3, the MIP model for scheduling wafer cleaning tools is developed in detail. Moreover, based on a backward strategy, we construct a TPN to model the behavior of the system to help determine the optimal tank assignment to the cleaning steps and the robot visiting order to the parallel tanks at each step. In Section 4, experimental results are used to validate the advantages of the proposed method compared to the traditional scheduling methods for wet stations. In Section 5, we present the summary of this research.

2. Literature review

In the semiconductor manufacturing industry, cluster tools are widely employed to perform critical wafer fabrication processes and they play a vital role in ensuring process efficiency and quality. Considering the substantial investment for these tools, it is necessary to operate them effectively to achieve higher productivity and utilization. Thus, great attention is given to the issues of scheduling cluster tools in semiconductor manufacturing. Typically, a cluster tool is equipped with two loadlocks (LLs) and thus two lots of wafers can be loaded into a tool concurrently. Hence, once one of the lots in one of LLs completes its processing, the tool can immediately switch to process the wafers of the other lot held in the other LL. As a result, a cluster tool can operate continuously under a steady-state for the majority of time, forming a repetitive cyclic production. Cluster tools are generally categorized into two types: Single-Arm Cluster Tools (SACT) and Dual-Arm Cluster Tools (DACT) according to the number of arms for the robot equipped in a tool. Extensive research has been dedicated to investigating optimal steady-state scheduling strategies for both types [9–14]. It has been demonstrated that, if the processing time of wafers on processing modules determines the productivity of a tool, the backward strategy yields the best performance for SACTs [15,16], while the swap strategy is proved to be optimal for DACTs [17,18]. In addition, given the increasingly strict requirements for wafer quality, incorporating residency time constraints (RTCs) into the scheduling of cluster tools is significant. In practical wafer fabrication, certain processes should comply with RTCs to ensure product quality, requiring that after a wafer is processed in a processing module (PM), it should be unloaded within a designated time window. Failing to do so may damage the wafer due to the excessive heat and lingering particles inside the PM.

With RTCs being taken into account for scheduling a cluster tool, it is crucial to minimize the delay time that a wafer experiences in a PM after its processing is finished. In the case of scheduling SACTs, this delay can be reduced by deferring the unloading action of a wafer at the previous processing step [19–21]. Moreover, in the case of scheduling DACTs, Kim et al. [22] demonstrated that increasing the robot's idle time during swap operations can shorten the wafer delay time. Rostami and Hamidzadeh [23] came up with a scheduling technique that combines linear programming with several heuristics, which can efficiently determine the optimal solution. Under the circumstances where the conventional swap strategy fails to meet RTCs for DACTs, in [24], it brings up a novel class of robot activity sequences designed to decrease wafer delay time. Roh et al. [25] analyzed the worst-case wafer delay in K-cyclic schedules for DACTs and provided operational insights to help minimize this upper bound. Moreover, for SACTs and DACTs, a feasible scheduling solution can be derived

using the feedback control approach introduced in [26–28]. Currently, with the advancement of technology, the demand for wafers from intelligent devices has become increasingly sophisticated and the operating scenarios of semiconductor equipment have grown more complex. Consequently, many researchers looking into cluster tool scheduling have begun to incorporate additional constraints into their models. Some researchers have focused on the scheduling of cluster tools equipped with chamber cleaning module [29–31], and some have been dedicated to the steady-state scheduling of multi-cluster tools [32–35]. Moreover, academic researchers have explored the scheduling problems of cluster tools that handle multiple wafer types [36–38].

Petri nets (PN) serve as an effective modeling tool for cluster tools, offering a more transparent depiction of wafer processing sequences and robotic arm operations. As a modeling framework for discrete event systems, PNs integrate graphical and mathematical representations [39]. Timed Petri Nets (TPN) extending conventional PNs by incorporating time delays in places and/or transitions have been generally utilized to model cluster tool scenarios, containing SACTs, DACTs, multi-cluster tools, and tools with wafer revisiting [40–42]. The researchers in [3] focused on cyclic scheduling problems in cluster tools with radial topologies. In their approach, TPNs were employed to model the system, followed by the development of a general mixed-integer programming (MIP) formulation to achieve a broad range of complex wafer scheduling tasks. Another notable PN variant is the resource-oriented Petri net (ROPN), which was generated by the researchers in [43,44]. The framework of ROPN can provide modeling flexibility and operate independently of specific wafer flow patterns. It is commonly applied to tackle scheduling challenges in cluster tools with various challenges such as RTCs [2,45], activity time variation [46,47], and transient processes [48,49]. In the current studies, Petri nets have been increasingly integrated with emerging techniques to enable iterative enhancements for addressing scheduling problems across a wide range of industrial domains. In [50], Petri nets were combined with anomaly recognition techniques, significantly improving the accuracy of abnormal event detection in cigarette manufacturing processes. Moreover, the researchers in [51] integrated deep reinforcement learning with topologically structured Petri nets by modeling them as graph convolutional networks, thereby enhancing their capability to respond to unexpected disturbances and enabling real-time scheduling.

Heuristic algorithms are also widely adopted in industrial manufacturing to address scheduling problems and constitute a mature research domain [52,53]. Commonly used heuristics for scheduling include Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA). These approaches have been also extensively applied to semiconductor equipment scheduling optimization. Qiao et al. [54] proposed an extended genetic algorithm integrated with Petri nets to improve scheduling strategies and achieve superior scheduling performance. In [55], PSO was applied to maintenance scheduling for semiconductor manufacturing equipment and further combined with simulated annealing. This hybrid approach effectively reduces maintenance time and mitigates downtime losses. Compared with MIP, heuristic algorithms typically yield solutions more rapidly, but they are prone to being trapped in local optima, which limits their effectiveness in significantly improving wet station operational efficiency. Moreover, in current manufacturing practice, the scale of wet stations and the complexity of wafer cleaning recipes are relatively moderate. As a result, the computational time required to obtain a global optimum using an MIP model remains well within acceptable limits. Therefore, we adopt an MIP-based modeling framework to achieve the proposed research objectives.

As mentioned, the linear structure of wet stations leads to distinct travel distances between pairs of tanks for a single-arm robot, which in turn considerably introduces additional complexity of their

scheduling tasks. Besides, compared to the abundant studies on cluster tool scheduling, investigations on cleaning tool scheduling remain relatively scarce. In [56], a set of efficient heuristics, combined with a tabu search procedure, were raised to get high-quality solutions within a reasonable computation time. Bhushan and Karimi [57] integrated a simulated annealing algorithm for sequencing the wafer jobs and an existing sequencing algorithm based on tabu search for wet station scheduling, which contributes to productivity improvement. Lee et al. [58] and Kim et al. [1] studied wet station scheduling problems involving multiple robotic arms through Petri net modeling. Lee et al. [58] addressed the problem using a mixed-integer programming formulation, Kim et al. [1] solved it via a branch and bound algorithm. Moreover, Aguirre et al. [59] established a mixed-integer programming formulation that attains a globally optimal schedule while reducing the number of tanks required in the tool. In their research, Novas and Henning [60] proposed an expressive constraint programming and afterward generalize this approach to minimize the unproductive intervals between the scheduling periods corresponding to varying wafer lot sets, while Pang et al. [7] developed a scatter simulated annealing algorithm for the scheduling problem of cleaning tools. Han et al. [8] integrated advanced machine learning techniques into wet station scheduling and develop an artificial intelligence-based scheduling framework capable of delivering practical and effective solutions in highly dynamic production environments. These researchers did not consider the two issues addressed in this work, which motivate us to conduct this study. Unlike other researchers, we take the different robot moving time for different tank pairs into account and attempt to improve the performance of a cleaning tool by optimizing the assignment of tanks to processing steps and the order of robot visiting the parallel tanks at each step.

3. Modeling for wet station

Before presenting the modeling process, we present the notations used in this paper. Table 1 summarizes all notations used in the formulation of the mathematical model and specifies their respective types for clarity. Notations classified as parameters require predefined values, whereas those identified as decision variables are endogenously determined by the model to achieve optimal solutions. Table 1 also provides concise definitions of each symbol to facilitate the interpretation and readability.

Table 1. Notations.

Symbol	Type	Meaning
T-i	Parameter	The i-th tank in a wet station.
m	Parameter	The number of tanks in a wet station.
n	Parameter	The number of cleaning steps in a wet station.
m_j	Parameter	The number of parallel tanks at Step j.
TK_{jk}	Parameter	The k-th ($k\in\{1, 2, \ldots, m_j\}$) tank serving for the j-th ($j\in\{1, 2, \ldots, n\}$) step.
TS_j	Parameter	The FOUP-visiting sequence of tanks at step j selected from the $m_j!$ options by a schedule.
TS_j^h	Parameter	The h-th sequence with which the m_j consecutive FOUPs visit the m_j parallel tanks for the j-th step.
S	Parameter	The least common multiple (LCM) of the n integers $m_1, m_2, \ldots, m_n$.

Continued on next page

Symbol	Type	Meaning
σ		The robot task sequence in a round.
σ_d		The robot task sequence in the *d*-th round.
r_j	Parameter	The number of FOUPs processed by each parallel tank in Step *j* during an S-FOUP cycle.
$L_{d,j}$		The loading task of the robot at Step *j* in the *d*-th round.
$U_{d,j}$		The unloading task of the robot at Step *j* in the *d*-th round.
λ	Parameter	The time required for both loading and unloading actions.
μ	Parameter	The robot moving time between two adjacent tanks.
$M_{(x,u)(y,v)}$		The robot moving from tank T-*x* allocated to step *u* to tank T-*y* allocated to step *v*, $x \neq y,\ u \neq v$.
δ	Parameter	The number of intermediate tanks between T-*x* and T-*y* (i.e., excluding T-*x* and T-*y*).
γ	Parameter	The distance from T-*x* to T-*y*.
α_j	Parameter	The processing time at Step *j*.
$RTS_{d,j}$		The robot activity sequence at Step *j* in the *d*-th round
END_d		A transition in the PN and it represents the completion of the robot task sequence σ for the *d*-th round in an *S*-FOUP cycle.
R_d		A timed place in the PN and it indicates the end of the *d*-th robot activity sequence and initiates the (*d*+1)-th one.
p	Parameter	The number of transitions in a round.
T	Decision Variable	The time span requested to finish *S* FOUPs and defined as the cycle time.
$X_{i,j}$	Decision Variable	A binary variable describes the tank allocation: $\forall i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$, if $X_{i,j} = 1$, it means that T-*i* is allocated to Step *j*, otherwise $X_{i,j} = 0$.
$a_{d,i,j}$1	Decision Variable	A binary variable describes the route selection: If tank T-*i* is to process an FOUP during the *d*-th round for step *j*, then $a_{d,i,j} = 1$, otherwise $a_{d,i,j} = 0$.
$Z(j, l)$	Decision Variable	A decision variable represents the tank index assigned to the *l*-th parallel tank of processing step *j* within the wet station.
$b_{j,k,l}$	Decision Variable	A binary variable introduced to reformulate the inequality constraints into a standard MIP formulation.
$\tau_{d,j}$1	Decision Variable	The wafer sojourn time that is spent in a tank at step *j* for the *d*-th round.
$t_{d,k}$1	Decision Variable	The starting time of firing the *k*-th transition in the *d*-th round, $k \in \{1, 2, \ldots, p\}$.
$Y_{d,j}$1	Decision Variable	The position of the tank that the robot moves to at step *j* in the *d*-th round.

3.1. Problem Description

In the research on scheduling wet stations, major efforts are made on cycle time reduction to enhance cleaning efficiency, but less attention is paid to the effect of robot traveling distances between tanks and the tank assignments to the processing steps on the performance of scheduling cleaning tools [7,8]. In other

studies, it is always assumed that the tanks are assigned to the processing steps according to the layout order in the tool. Specifically, operators dispense the corresponding cleaning solutions sequentially from left to right, following the order of the cleaning steps, from the first to the final stage, and the arrangement of parallel tanks within each step. For example, if there are four tanks in a tool and they are labeled T1, T2, T3, and T4, then for a two-step cleaning process with the two steps requiring three and one tanks, respectively, it assigns tanks T1, T2, and T3 to step 1, and T4 for step 2. However, in actual production, the tank can be assigned in different ways, and the performance may be improved by optimally assigning them. For instance, for the above example, one can assign T1, T3, and T4 to step 1, and T2 to step 2. Thus, it is meaningful to examine whether the change of tank assignment to steps has affects the performance of a wet station, which is one of our focuses of this study. If yes, given a cleaning process, it is meaningful to decide the optimal tank configuration.

With the linear configuration of wet stations, tank assignment to steps is not the only issue worth investigating in terms of scheduling optimization. The order for the robot to visit the parallel tanks in a step may affect the performance of the cleaning tool, since the different robot visiting order to these tanks results in different robot movement time between tanks, which is our other focus of this study. For instance, in a wet station consisting of three tanks for a two-step cleaning process, Tanks 1 and 2 are designated to Step 1, while Tank 3 is assigned to Step 2. In general, after completing Step 1 in Tank 1, the first FOUP is then transferred to Tank 3 for Step 2, while the second FOUP is processed by Tank 2 and then Tank 3. Nevertheless, owing to the tool's linear structure, the distances from Tanks 1 and 2 to Tank 3 are different. Consequently, in terms of productivity, it may be better if the first FOUP is processed by Tank 2, while the second FOUP by Tank 1, which is dependent on the processing parameters. If this holds true, the wafer placement sequence to the parallel tanks in a step should be optimized to minimize the overall cycle time.

Because certain cleaning steps require corrosive acidic solutions, wafer residency time constraints constitute a critical consideration in semiconductor equipment scheduling and are frequently addressed in wafer cleaning processes. However, the wet station investigated in this study employs a batch wafer cleaning technology designed for high-volume manufacturing. Compared with single wafer cleaning, batch processing imposes less stringent precision requirements. Consequently, the residency time allocated to each cleaning step is generally sufficiently long. In typical semiconductor wet cleaning processes, the allowable residency time window for batch cleaning is on the order of several minutes; for example, common cleaning steps such as Piranha or IMEC cleans typically have a processing time ranging from 2 to 10 minutes, with permissible overstay margins often reaching 2–3 minutes. This time buffer is more than adequate to safely prevent any risk of wafer scrapping due to overstaying in chemical tanks. Under these conditions, the residency time constraint can be reasonably omitted in this study. Furthermore, in semiconductor equipment scheduling, two typical scenarios are commonly encountered: Cases in which the robot has sufficient idle time and cases in which it does not. When adequate waiting time is available, appropriate scheduling can ensure that wafers are unloaded immediately upon process completion, thereby eliminating the concerns regarding residency time violations. In contrast, when the robot's waiting time is insufficient, wafers may be forced to remain in certain tanks after processing, and the resulting additional waiting time must be strictly controlled within the allowable residency limits. In this research, by jointly optimizing the tank layout and the robot visiting sequence, the robot's travel time is significantly reduced, enabling timelier wafer retrieval and thereby facilitating the compliance with wafer residency time constraints.

For a wet station with m tanks that are linearly arranged, we label them according to the arranging order, i.e., the i-th tank is labeled T-i. The serial-parallel flow pattern for wafers to be processed is denoted as $(m_1, m_2, \ldots, m_n)$, with n being the number of steps and m_j as the number of parallel tanks required for the j-th step. Under this circumstance, we get

$$\sum_{j=1}^{n}(m_j) = m \tag{1}$$

In this study, we treat the storages of raw and completed wafers as Step 0 and Step n+1, and the input buffer (labeled T-0) and output buffer (labeled T-(m+1)) serve them, respectively. We assume that T-0 and T-(m+1) have infinite capacity, while a tank can accommodate a single FOUP only; i.e., the capacity is one. Thus, with the wafer flow pattern $(m_1, m_2, \ldots, m_n)$, a raw FOUP is taken from Step 0, then goes through Steps 1 to n, and is finally transferred to Step n+1 to complete the cleaning process.

We use TK_{jk} to denote the k-th ($k \in \{1, 2, \ldots, m_j\}$) tank serving for the j-th ($j \in \{1, 2, \ldots, n\}$) step. With multiple parallel tanks for a step, FOUPs can visit these tanks in different order. We use TS_j^h to denote the h-th ($h \in \{1, 2, \ldots, m_j!\}$) access sequence with which the m_j consecutive FOUPs visit the m_j parallel tanks for the j-th step. Under this setting, when the robot visits processing step j, it can choose one of $m_j!$ feasible permutations to visit the corresponding parallel tanks. For example, if $m_j = 3$, there are totally $m_j! = 3! = 6$ optional visiting sequences including $TS_j^1 = (TK_{j1}, TK_{j2}, TK_{j3})$, $TS_j^2 = (TK_{j1}, TK_{j3}, TK_{j2})$, $TS_j^3 = (TK_{j2}, TK_{j1}, TK_{j3})$, $TS_j^4 = (TK_{j2}, TK_{j3}, TK_{j1})$, $TS_j^5 = (TK_{j3}, TK_{j1}, TK_{j2})$, and $TS_j^6 = (TK_{j3}, TK_{j2}, TK_{j1})$. Besides, if $m_j = 4$, there are totally $m_j! = 4! = 24$ optional sequences such as $TS_j^1 = (TK_{j1}, TK_{j2}, TK_{j3}, TK_{j4})$, $TS_j^2 = (TK_{j1}, TK_{j2}, TK_{j4}, TK_{j3})$, $TS_j^3 = (TK_{j1}, TK_{j3}, TK_{j2}, TK_{j4})$, and so on. Note that with different TS_j^h and TS_{j+1}^d, an FOUP visits different tanks at steps j and j+1, resulting in different robot moving distance and different moving time. Thus, TS_j^h should be taken into account in scheduling a cleaning tool.

To facilitate more effective cleaning tool scheduling, the least common multiple (LCM) is introduced for analyzing wet station scheduling. Given a wafer flow pattern $(m_1, m_2, \ldots, m_n)$, we use S to denote the least common multiple (LCM) of the n integers $m_1, m_2, \ldots, m_n$. As discussed above, the visiting sequence of parallel tanks by FOUPs for step j may affect the performance in scheduling a clean station. Since our objective of this study is to maximize the cleaning efficiency of cleaning tools, the system is assumed to operate under full-load conditions at steady state, implying that all parallel tanks at each processing step are continuously active. Under this regime, the robot follows a predetermined access sequence to load and unload FOUPs at each parallel tank of step j until all tanks are occupied and the cleaning operations are initiated. The robot then returns to the first visited tank and repeats the same visiting order to start the next cycle. Once the visiting sequence is determined in the initial cycle, it remains fixed in the subsequent cycles. Any deviation from this sequence would force additional waiting for wafer retrieval at certain parallel tanks, thereby increasing the overall cycle time and prolonging wafer residency time in some parallel tanks, which is undesirable. In addition, the determination of the robot access order is incorporated into the optimization framework through appropriate decision variables and constraints, enabling the model to identify the optimal visiting sequence for the robot.

We let TS_j denote the FOUP-visiting sequence of tanks at step j selected from the $m_j!$ options by a schedule and $[TK_{jg}, TK_{(j+1)h}]$ denote that an FOUP visits TK_{jg} at step j and then visits $TK_{(j+1)h}$ at step (j+1). For instance, with wafer flow pattern (2, 4), if a schedule selects $TS_1 =$

(TK_{11}, TK_{12}) and $TS_2 = (TK_{21}, TK_{22}, TK_{23}, TK_{24})$ as the FOUP-visiting sequences for steps 1 and 2, respectively, then the route for the five consecutive FOUPs to visit the tanks at steps 1 and 2 can be described as $\langle [TK_{11}, TK_{21}] \rightarrow [TK_{12}, TK_{22}] \rightarrow [TK_{11}, TK_{23}] \rightarrow [TK_{12}, TK_{24}] \rightarrow [TK_{11}, TK_{21}] \rangle$. It is worth noting that the 1st and 5th FOUPs visit the same pair of tanks from Step 1 and Step 2, and then repeats the same route. This means that once four FOUPs have been finished, the system returns to the same state, thereby forming a periodic cycle. In this illustrative case, we can find that $S = 4$; i.e., the LCM of 2 and 4. Furthermore, an important observation is that the cycle remains constant, regardless of the wafer flow pattern and the robot access sequence adopted at any step. With each such a cycle, *S* FOUPs are processed, yielding an *S*-FOUP cyclic schedule. The time span requested to finish *S* FOUPs is defined as the cycle time and represented by *T*.

In a cycle of an *S*-FOUP schedule, a round of robot actions (a round in short) describes the robot task sequence σ = ⟨picks an FOUP from step n → moves to the output buffer → places the FOUP there → moves to step n-1 → picks an FOUP from there → moves to step n → places the FOUP there → moves to step n-2 → … → moves to the input buffer → picks an FOUP from there → moves to step 1 → places the FOUP there → moves to step n⟩ that is repeated periodically. During a cycle, the robot repeats *S* rounds and *S* FOUPs of wafers are processed at each step. Hence, a parallel tank at Step *j* handles r_j FOUPs calculated by

$$r_j = \frac{S}{m_j} \tag{2}$$

Our goal of scheduling a cleaning tool is to maximize the operating efficiency, i.e., to minimize the cycle time of the *S*-FOUP cyclic schedule. Thus, the objective for the MIP model is given as

$$\min (T) \tag{3}$$

Then, we present the decision variables and constraints to complete the formulation.

3.2. Discrete Decision Variables

3.2.1. Tank Allocation Variables and Constraints

We have discussed the effect of the FOUP-visiting sequence to the parallel tanks at each processing step on the robot moving distance in a schedule. Here, we examine the effect of the assignment of tanks to steps with respect to the robot moving distance in a schedule. A practical example is provided to illustrate this issue. Assume that the wet station has three tanks that are labeled T-1, T-2, and T-3, and the wafer flow pattern is (2, 1). Then, we have S = 2, and a 2-FOUP cycle schedule can be obtained. We can set TK_{11} = T-1, TK_{12} = T-2, and TK_{21} = T-2 to get Assignment 1: (TK_{11}, TK_{12}, TK_{21}) = (T-1, T-2, T-3). Alternatively, we can get Assignment 2 as (TK_{11}, TK_{12}, TK_{21}) = (T-1, T-3, T-2). Then we select the FOUP-visiting sequence for both assignments as $TS_1 = (TK_{11}, TK_{12})$ and $TS_2 = (TK_{21})$. For assignment (TK_{11}, TK_{12}, TK_{21}) = (T-1, T-2, T-3), the robot task sequence of the 1st round is ⟨T-3 (TK_{21}) → T-4 (output buffer) → T-1 (TK_{11}) → T-3 (TK_{21}) → T-0 (input buffer) → T-1 (TK_{11}) → T-3 (TK_{21})⟩, and the robot task sequence of the 2nd round is ⟨T-3 (TK_{21}) → T-4 (output buffer) → T-2 (TK_{12}) → T-3 (TK_{21}) → T-0 (input buffer) → T-2 (TK_{12})→ T-3 (TK_{21})⟩. For assignment (TK_{11}, TK_{12}, TK_{21}) = (T-1, T-3, T-2), the robot task sequence of the 1st round is ⟨T-2 (TK_{21}) → T-4 (output buffer) → T-1 (TK_{11}) → T-2 (TK_{21}) → T-0 (input buffer) → T-1 (TK_{11}) → T-2 (TK_{21})⟩, and the robot task sequence of the 2nd round is ⟨T-2 (TK_{21}) → T-4 (output buffer) →

T-3 (TK_{12}) → T-2 (TK_{21}) → T-0 (input buffer) → T-3 (TK_{12}) → T-2 (TK_{21})⟩. To calculate the total robot traveling distance, it is reasonable to assume that the distance between two adjacent tanks is one unit. Then, for assignment (TK_{11}, TK_{12}, TK_{21}) = (T-1, T-2, T-3), the robot travels 22 units of distance in a cycle; while for assignment (TK_{11}, TK_{12}, TK_{21}) = (T-1, T-3, T-2), it travels 20 units of distance in a cycle. Hence, assignment (TK_{11}, TK_{12}, TK_{21}) = (T-1, T-3, T-2) can save the robot moving time. This observation leads to the conclusion that the assignment of tanks to processing steps can directly affect the operational efficiency of a wet station.

Thus, to formulate an MIP for the problem, we use a binary variable $X_{i,j}$ to describe the tank allocation: $\forall i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$, if $X_{i,j} = 1$, it means that T-i is allocated to Step j, otherwise, $X_{i,j} = 0$. Since a tank can be allocated to a single step only, we have the following constraint:

$$\sum_{j=1}^{n}(X_{i,j}) = 1, \quad i \in \{1, 2, \ldots, m\} \tag{4}$$

Besides, for step j, it should configure exactly m_j tanks such that the following constraint should hold.

$$\sum_{i=1}^{m}(X_{i,j}) = m_{j,} \quad j \in \{1, 2, \ldots, n\} \tag{5}$$

3.2.2. Route Selection Variables and Constraints

Similarly, as discussed, the order for the robot to visit the parallel tanks in scheduling a wet station affects the performance of the tool. To illustrate how such an order affects the performance, a simple example is presented as follows: For this example, the wet station is configured with six tanks with wafer flow pattern being (4, 2). Since S = 4, in this case, a 4-FOUP cyclic schedule will be generated such that in a cycle each tank at step 1 processes one FOUP, while each tank at step 2 processes two FOUPs. Assume that the tank allocation is known as (TK_{11}, TK_{12}, TK_{13}, TK_{14}, TK_{21}, TK_{22}) = (T-1, T-2, T-3, T-4, T-5, T-6). We examine different sequences for the robot to visit the parallel tanks at the two steps. For the first sequence, we set $TS_1 = (TK_{11}, TK_{12}, TK_{13}, TK_{14})$ and $TS_2 = (TK_{21}, TK_{22})$, whereas for the second, we set $TS_1 = (TK_{11}, TK_{12}, TK_{14}, TK_{13})$ and $TS_2 = (TK_{21}, TK_{22})$. Then, for the first sequence, in a cycle, the processing routes of the two FOUPs that go through T-5 (TK_{21}) at Step 2 are ⟨T-5 → T-7 (output buffer) → T-1 → T-5⟩ and ⟨T-5 → T-7 (output buffer) → T-3 → T-5⟩, respectively, implying that the robot moves 20 units of distance, while the processing routes of the other two FOUPs that go through T-6 (TK_{22}) at Step 2 are ⟨T-6 → T-7 (output buffer) → T-2 → T-6⟩ and ⟨T-6 → T-7 (output buffer) → T-4 → T-6⟩, respectively, resulting in that the robot movement distance is 16 units. Note that the robot movements between the FOUPs that go through T-5 and T-6 are not balanced, and this affects the performance of the tool. On the contrary, for the second sequence, the processing routes of the two FOUPs that go through T-5 (TK_{21}) are ⟨T-5 → T-7 (output buffer) → T-1 → T-5⟩ and ⟨T-5 → T-7 (output buffer) → T-4 → T-5⟩, respectively, resulting in that the robot movement distance through T-5 is 18 units; while the processing routes of the other two FOUPs that go through T-6 (TK_{22}) are ⟨T-6 → T-7 (output buffer) → T-2 → T-6⟩ and ⟨T-6 → T-7 (output buffer) → T-3 → T-6⟩, respectively, resulting in 18 units of the robot movement distance. In this case, the robot movement distances between the FOUPs through T-5 and T-6 are balanced, which may improve the performance obtained by the first sequence. Therefore, in this work, we also attempt

to optimize the order for the robot to visit the parallel tanks at the processing steps to improve the performance in scheduling a wet station.

In the MIP model, to select an optimal order for the robot to visit the parallel tanks at the processing steps, a binary variable $a_{d,i,j}$ is defined as follows: If tank T-*i* is to process an FOUP during the *d*-th round for step *j*, then $a_{d,i,j} = 1$, otherwise $a_{d,i,j} = 0$. Notice that, for each round, a tank at a step can process an FOUP only. Hence, Constraint (6) given below should be satisfied.

$$\sum_{i=1}^{m} a_{d,i,j} = 1, \qquad d \in \{1, 2, \ldots, S\} \text{ and } j \in \{1, 2, \ldots, n\} \tag{6}$$

Second, for any cleaning step in the *d*-th round, this selection should be consistent with the tank allocation, and we have

$$a_{d,i,j} \leq X_{i,j}, \qquad d \in \{1, 2, \ldots, S\} \text{ and } i \in \{1, 2, \ldots, m\} \text{ and } j \in \{1, 2, \ldots, n\} \tag{7}$$

It guarantees that, in the *d*-th round of robot activities, the FOUP to be performed at Step *j* is precisely delivered to the tank that is designated for Step *j*, preventing tanks that are assigned to other steps from being incorrectly utilized for the execution of Step *j*. Last, throughout the *S*-FOUP cycle, the following constraint should be fulfilled:

$$\sum_{d=1}^{S} a_{d,i,j} = r_j \times X_{i,j,} \qquad i \in \{1, 2, \ldots, m\} \text{ and } j \in \{1, 2, \ldots, n\} \tag{8}$$

This means that each tank within the cleaning tool is subject to a usage constraint. In particular, every parallel tank allocated to Step *j* should perform the corresponding cleaning operation exactly r_j times within an *S*-FOUP cycle. This constraint ensures high utilization and operational stability of a wet station.

3.2.3. Variables to Ensure the Selected Visiting Order and Constraints

In practice, to optimize the overall operational efficiency of a wet station, with flow pattern $(m_1, m_2, \ldots, m_n)$, under a steady state, all the $\sum_{j=1}^{n}(m_j) = m$ tanks are in operation. Under steady state, we analyze how to describe the robot visiting order to the tanks at a step. For instance, a wet station contains five tanks and the wafer flow pattern is (3, 2). Since $S = 6$, the robot executes six rounds of activity sequences in a cycle. In this case, to increase the productivity of the cleaning tool, for Step 1, the robot should visit the same parallel tank, such as TK_{11} in the 1st and 4th rounds, then access another parallel tank, like TK_{12} in the 2nd and 5th rounds, and finally visit the remaining parallel, tank like TK_{13} in the 3rd and 6th rounds. Similarly, for Step 2, the robot is required to visit the identical parallel tank, such as TK_{21} in the 1st, 3rd, and 5th rounds, and access the other parallel tank, such as TK_{22} in the 2nd, 4th, and 6th rounds. Nevertheless, the variables and constraints defined so far are insufficient for enabling the robot to accomplish the above tasks. Without adding essential additional constraints, in the aforementioned example, for Step 1, it is likely that the robot visits TK_{11} in the 1st and 2nd rounds, accesses TK_{12} in the 3rd and 4th rounds, and visits TK_{13} in the 5th and 6th rounds. Such a scenario is undesirable because it conflicts with the goal of efficiency optimization. Therefore, it is further necessary to incorporate relative variables and corresponding constraints into the MIP model, which can also decrease the computational time for achieving the optimal solution.

To implement the above requirement, among the m_j consecutive FOUPs to be processed, each of them should visit only one of the tanks TK_{j1}, TK_{j2}, …, TK_{jm_j} in a selected order. We set $Z(j, l) \in \{1, 2, \ldots, m_j\}$ as a decision variable and $l \in \{1, 2, \ldots, m_j\}$ to denote that, among the m_j consecutive

FOUPs, the $Z(j, l)$-th FOUP should visit tank TK_{jl} at step j. In this way, the order for the robot to visit the allocated tanks is determined. With these variables, we first have

$$X_{Z(j,l),j} = 1, \quad j \in \{1, 2, \ldots, n\} \text{ and } l \in \{1, 2, \ldots, m_j\} \tag{9}$$

This implies that the parallel tanks $TK_{j\,l}{}'s$ assigned to Step j exclusively perform the cleaning activity corresponding to that step. In addition, these variables are supposed to fulfill, i.e., $Z(j,1) \neq Z(j,2) \neq Z(j,3) \neq \cdots \neq Z(j,m_j), j \in \{1, 2, \ldots, n\}$. However, based on the MIP formulations, the preceding inequality constraints should be reformulated by introducing a binary variable $b_{j,k,l}$ together with the Big-M method, as follows:

$$Z(j,k) \leq Z(j,l) - \varepsilon + M \times (1 - b_{j,k,l}), \quad Z(j,k) \geq Z(j,l) + \varepsilon - M \times b_{j,k,l},$$

$$j \in \{1, 2, \ldots, n\} \text{ and } k, l \in \{1, 2, \ldots, m_j\} \text{ and } k < l \tag{10}$$

where ε denotes a sufficiently small positive constant and M represents a sufficiently large positive constant. This indicates that each parallel tank $TK_{j\,l}$ allocated to Step j is mapped to a unique tank index to determine the robot visiting order. Finally, for any parallel tank TK_{jl} allocated to Step j, for the d-th round with $d = m_j \times (q-1) + l,\ q \in \{1, 2, \ldots, r_j\}$, it is necessary to hold the following constraint:

$$a_{d,Z(j,l),j} = 1, \quad j \in \{1, 2, \ldots, n\} \text{ and } l \in \{1, 2, \ldots, m_j\} \tag{11}$$

Constraint (11) determines the order with which the multiple parallel tanks assigned to Step j are visited by the robot in a cyclic way throughout the S-FOUP cycle. Thus, the MIP model optimizes the order for the robot to visit the parallel tanks at step j.

3.3. Continuous Decision Variables

3.3.1. Backward Strategy

During the cyclic operation of a wet station, the single-arm robot plays a pivotal role in wafer transportation and its tasks primarily consist of three actions: FOUP loading (loading in short), FOUP unloading (unloading in short), and moving. Loading is defined as the task that an FOUP of wafers is placed into a tank by the robot, while unloading is referred as to the task that an FOUP of wafers is picked up from a tank (or a buffer) by the robot. In this context, we utilize $L_{d,j}$ and $U_{d,j}$ to represent the loading and unloading tasks, respectively, at Step j in the d-th round. The time required for both actions are assumed identical, symbolized by λ, and this duration is substantially shorter than the processing time demanded by wafer cleaning in a tank. For robot moving, within this work, the robot moving time between two adjacent tanks is treated as a given constant, expressed as μ. In practical linear rail operations, the robot is typically controlled using a trapezoidal velocity profile (rapid acceleration–constant velocity–rapid deceleration) to maximize throughput. Under this control strategy, the acceleration and deceleration phases are very short, often on the order of a few tenths of a second, causing the constant-velocity phase to dominate the overall motion. Consequently, the robot's moving time exhibits a strong linear relationship with the travel distance. Therefore, the robot's traveling time in the wet station can be reasonably modeled as a linear function. We use $M_{(x,u)(y,v)}$ to denote the

robot moving from tank T-x allocated to step u to tank T-y allocated to step v, $x \neq y$ and $u \neq v$. Assume that, between T-x and T-y, there are δ intermediate tanks (i.e., excluding T-x and T-y), then the distance from T-x to T-y is formally defined as $\gamma = \delta + 1$, and the time taken for the robot to move from T-x to T-y is $\gamma \times \mu$. For instance, between tanks T-x and T-y, if there are $\delta = 2$ tanks between them and the distance between them is $2 + 1 = 3$, thus the robot's moving time between them is 3μ. With the linear configuration of the tanks in a clean station, although the task sequence σ repeated by the robot remains identical for every round, the tanks visited by the robot during σ for different rounds vary. As a result, the robot's traveling time is not constant but varies from one round to another.

Besides the robot activity time, the model also needs to describe the wafer processing time. The time during which an FOUP stays in a tank is referred to as wafer residency time, and $\tau_{d,j}$ is used to denote the wafer sojourn time that is spent in a tank at step j for the d-th round. At every cleaning step, parallel tanks are requested to comply with the First-In-First-Out (FIFO) rule, which means that when one of the parallel tanks assigned to a certain processing step initiates wafer cleaning earlier than the others, the robot is obliged to unload wafers from that tank first. To complete the processing of an FOUP of wafers, an FOUP should stay in a tank at step j α_j time units. This is called the processing time, and we have that $\tau_{d,j} \geq \alpha_j$ should hold.

To maximize the production of a cleaning tool under steady-state, all the tanks in the system need to operate concurrently, which is governed by the backward strategy since a single-arm robot is equipped. By adopting the backward strategy and FIFO rule for the parallel tanks at a step, the robot activity sequence at Step j in the d-th round is represented as $RTS_{d,j} = \langle U_{d,j} \rightarrow M_{(x,j)(y,j+1)} \rightarrow L_{d,j+1} \rangle$. Thus, without robot waiting, in the d-th round, the task sequence performed by the robot is $\sigma_d = \langle RTS_{d,n} \rightarrow M_{(x,n+1)(y,n-1)} \rightarrow RTS_{d,n-1} \rightarrow M_{(x,n)(y,n-2)} \rightarrow \ldots \rightarrow RTS_{d,1} \rightarrow M_{(x,2)(y,0)} \rightarrow RTS_{d,0} \rightarrow M_{(x,1)(y,n)} \rangle$. Then, $\sigma_1, \sigma_2, \ldots, and\ \sigma_s$ together form a robot cycle with cycle time T, and S FOUPs of wafers are completed and delivered to the output buffer. Such a process is repeated to constitute a cyclic schedule.

3.3.2. Timed Petri Net for the Process

With the above discussion, to complete the MIP formation, timed Petri net is adopted to model the behavior of the system. By constructing a Timed Petri net (TPN), the robot action sequence throughout the S-FOUP cycle is presented in a more intuitive and transparent manner. Leveraging fundamental mathematical concepts such as states, events, concurrency, and synchronization, TPN provides an unambiguous formal semantic model of the robot task sequence. This ensures that the resulting diagram itself constitutes a logically rigorous mathematical object, which facilitates the systematic definition of decision variables and constraints in subsequent modeling stages. Notably, the Petri net serves as a structural and semantic foundation for the model rather than a computational mechanism, and its incidence matrices are not required for deriving the optimal solution. PNs constitute a special class of directed graphs composed of places and transitions. In graphical representation, places are depicted as circles ("◯"), while transitions are described as black rectangles ("▬"). The interconnections between places and transitions are expressed through directed arcs. The PN model that describes the robot task sequence is illustrated in Figure 2.

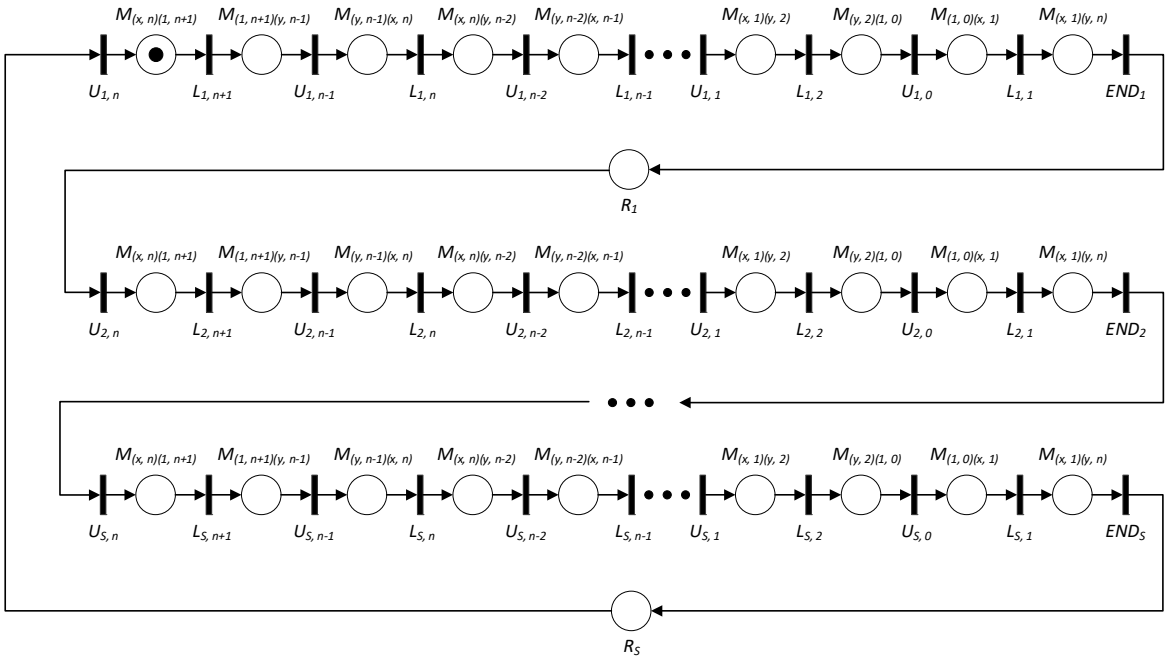

Figure 2. The PN model of a cleaning station in a cycle.

In this model, the robot unloading $U_{d,j}$ and loading $L_{d,j}$ are modeled as transitions and the time taken for the firing of such a transition takes λ time units. The robot moving $M_{(x,u)(y,v)}$ is modeled by a timed place. When a token enters such a place, it must reside there for some time so that the robot movement can finish, and is dependent on the travelling distance. Notice that a token may stay in a place longer than that required by the movement since it needs to wait there for some time. Furthermore, transition END_d represents the completion of the robot task sequence σ for the *d*-th round in an *S*-FOUP cycle, and it should be an instantaneous transition. The firing of $U_{d,n}$ indicates the removing of a completed FOUP from a tank at Step *n*, thereby initiating the activity sequence for the *d*-th round. Timed place R_d indicates the end of the *d*-th robot activity sequence and initiates (*d*+1)-th. When a token enters this place, it can immediately proceed to the subsequent place without incurring any time delay. From this model, one can observe that once a token enters $U_{1,n}$, it traverses these places sequentially according to the transition order in the model and comes back to $U_{1,n}$, completing an *S*-FOUP cycle in a wet station with cycle time *T*. By repeating this cycle, a stable periodic operation is achieved.

3.3.3. Time Variables and Constraints

Based on the constructed Petri net model, we incorporate the time decision variables and corresponding constraints into the MIP formulation. Note that there are $p = 2n + 3$ transitions in a round and they fire sequentially with each firing only once in a cycle. We define $t_{d,k}$ as the starting time of firing the *k*th transition in the *d*th round, $k \in \{1, 2, \ldots, p\}$. Thus, to determine an optimal schedule, it is used to determine the decision variables $t_{d,k}$'s. We then discuss how to determine them by presenting relevant constraints.

First, consider a timed place $M_{(x,u)(y,v)}$ in a given round in the PN model and assume that its input and output transitions are u_1 and u_2, respectively. Further, assume that u_1 and u_2 begin to fire at t_1 and t_2, respectively. Then, $t_2 - t_1 =$ the time taken for firing u_1 + the robot moving time

taken for $M_{(x,u)(y,v)}$. It is known that it takes λ time units for firing u_1, while the time for firing $M_{(x,u)(y,v)}$ is dependent on the robot moving distance, and we need to know the accurate positions of the two tanks between which the robot travels. To do so, by $M_{(x,u)(y,v)}$, the robot is to move to a tank at step *j* in the *d*-th round, and we use $Y_{d,j}$ to denote the position of this tank. In this setting, the input buffer T-0 for step 0 is fixed at position 0, i.e., $Y_{d,0} = 0$; while the output buffer T-(*m*+1) for Step *n*+1 is fixed at position *m*+1, i.e., $Y_{d,n+1} = m + 1$. The position of a tank visited by the robot in the *d*-th round at Step *j* is then calculated according to the following formulation:

$$Y_{d,j} = \sum_{i=1}^{m}\left(a_{d,i,j} \times i\right), \qquad d \in \{1, 2, \ldots, S\} \text{ and } j \in \{1, 2, \ldots, n\} \tag{12}$$

With $Y_{d,j}$ known, we can calculate the distance between the two tanks as follows: Assume that the robot movement is from steps *u* to *v* in the *d*-th round, then both $Y_{d,u}$ and $Y_{d,v}$ are known, and we can calculate the distance between them, which is denoted as $|Y_{d,u} - Y_{d,v}|$. Note that we have $|Y_{d,u} - Y_{d,v}| = |Y_{d,v} - Y_{d,u}|$. Then, the time taken for this robot movement is $|Y_{d,u} - Y_{d,v}| \times \mu$. For instance, in the 1st round, the time required for the robot to move from Step 1 to Step 2 is given by $|Y_{1,2} - Y_{1,1}| \times \mu$. Then, with the PN model, to determine $t_{d,k}$'s, we have the following constraint:

$$t_{d,k+1} - t_{d,k} \geq \lambda + \left|Y_{d,u} - Y_{d,v}\right| \times \mu, \qquad d \in \{1, 2, \ldots, S\} \text{ and } k \in \{1, 2, \ldots, p-1\} \tag{13}$$

As discussed, in the *d*-th round, an FOUP should stay in a tank at step *j* for $\tau_{d,j} \geq \alpha_j$ time units to complete the processing of wafers. With the PN model, it can be observed that an FOUP that is loaded into a tank at Step *j* in the *d*-th round is unloaded in the $(d + m_j)$-th round. Furthermore, loading an FOUP into a tank at step *j* is performed by firing the *k*-th transition in the *d*-th round with $k = 2(n - j) + 4$, while unloading the FOUP is performed by firing the *h*-th transition in the $(d + m_j)$-th round with $h = 2(n - j) + 1$. Thus, we have $\tau_{d,j} = t_{d+m_j,2(n-j)+1} - t_{d,2(n-j)+4} - \lambda$, and the following constraint should be satisfied in the MIP model.

$$t_{d+m_j,2(n-j)+1} - t_{d,2(n-j)+4} - \lambda \geq \alpha_j, \qquad d \in \{1, 2, \ldots, S\} \text{ and } j \in \{1, 2, \ldots, n\} \tag{14}$$

In a cycle, the robot performs $\sigma_1, \sigma_2, \ldots, and\ \sigma_s$ sequentially, implying that σ_{d+1} can begin only after σ_d is completed. Therefore, Constraint (15) given below should be imposed.

$$t_{d+1,1} \geq t_{d,p}, \qquad d \in \{1, 2, \ldots, S\} \tag{15}$$

The PN model given in Figure 2 describes the process of a cycle. After a cycle ends, another cycle begins, and each cycle should have the same cycle time. Note that $t_{d,k}$ denotes the time for firing the *k*-th transition in the *d*-th round of a cycle and $t_{d+S,k}$ should be the time for firing the *k*-th transition in the *d*-th round of the next cycle. Since the cycle time for each cycle is identical, the following constraint should be satisfied:

$$t_{d,k} + T = t_{d+S,k}, \qquad d \in \{1, 2, \ldots, S\} \text{ and } k \in \{1, 2, \ldots, p\} \tag{16}$$

As shown in the PN model, $t_{S,p}$ denotes that the first cycle is completed, which must equal the cycle time. Thus, we have Constraint (17) given below:

$$T = t_{S,p} \tag{17}$$

Up to now, we have presented the MIP model for finding an optimal cyclic schedule for a wet station by optimizing the allocation of tanks to processing steps and the robot visiting order to the parallel tanks at each step. Thus, we have presented the objective and all the constraints.

4. Experimental results

After establishing the MIP model, we proceed to construct experimental scenarios and employ a solver to obtain the optimal solutions for each scenario, thereby reporting the effect of optimizing the problem addressed in this work, validating the effectiveness of the proposed model, and comparing its performance with that of conventional sequential wet station scheduling strategies. In this study, experiments are conducted using an AMD Ryzen 7 PRO 6850U with Radeon Graphics 2.69 GHz CPU, 16.0 GB RAM to evaluate the performance of the approach. The model is implemented in Python and solved using the GUROBI 10.0.2 optimizer. The results are compared with the method that allocates the tanks to processing steps and visits the parallel tanks at a step according to the order of the tank arrangement in the tool. Specifically, the compared experimental configuration adopts a conventional tank layout in which tanks are arranged from left to right according to the processing sequence of cleaning steps. Correspondingly, when transporting an FOUP to a given step, the robot visits the parallel tanks in a left-to-right order by default. This layout and access policy are widely regarded as canonical in industrial wet cleaning tools and are commonly assumed in studies on wet station scheduling [57]. For instance, assume that the wafer flow pattern is (2, 3, 2), which makes the tank allocation (TK_{11}, TK_{12}, TK_{21}, TK_{22}, TK_{23}, TK_{31}, TK_{32}) = (T-1, T-2, T-3, T-4, T-5, T-6, T-7), and the order for the robot to visit the parallel tanks at the step is set as TS_1 = (TK_{11}, TK_{12}), TS_2 = (TK_{21}, TK_{22}, TK_{23}), and TS_3 = (TK_{31}, TK_{32}). Moreover, the baseline wet station scheduling also employs a backward strategy. Under steady-state cyclic operation, all parallel tanks at every cleaning step are continuously active, so that the wet station operates under a fully loaded regime, thereby achieving maximal cleaning efficiency. The experimental data and results are reported in Table 2.

In this table, the first column lists the experiment indices, while the second column displays the wafer flow patterns for each experiment. The third column provides the basic parameter settings, including the processing time of individual cleaning steps, the robot's loading and unloading time, and the robot's moving time between adjacent tanks. The fourth column reports the optimal cycle time obtained from the proposed MIP model under the specified wafer flow pattern and parameter settings. For comparison, the fifth column presents the optimal cycle time derived from the compared method. Finally, the last column records the computational time required by GUROBI to solve the proposed MIP model and obtain the optimal wet station cyclic schedule across experimental settings.

Table 2. Scheduling Results of Wet Stations.

No.	Wafer flow pattern	Parameter Setting	Optimal Cycle	The compared method	Computational Time (s)
1	(2, 1)	$\lambda = 10, \mu = 5, \alpha_1 = 300, \alpha_2 = 200$	530	530	0.007
2	(2, 1)	$\lambda = 20, \mu = 15, \alpha_1 = 180, \alpha_2 = 100$	540	570	0.006
3	(3, 2)	$\lambda = 15, \mu = 5, \alpha_1 = 400, \alpha_2 = 250$	1050	1050	0.024
4	(3, 2)	$\lambda = 15, \mu = 10, \alpha_1 = 300, \alpha_2 = 200$	1460	1560	0.025
5	(4, 2)	$\lambda = 5, \mu = 10, \alpha_1 = 440, \alpha_2 = 330$	880	900	0.019
6	(4, 2)	$\lambda = 10, \mu = 10, \alpha_1 = 150, \alpha_2 = 200$	960	1040	0.044
7	(1, 2, 2)	$\lambda = 10, \mu = 5, \alpha_1 = 200, \alpha_2 = 300, \alpha_3 = 150$	530	530	0.007
8	(1, 2, 2)	$\lambda = 5, \mu = 15, \alpha_1 = 160, \alpha_2 = 600, \alpha_3 = 460$	680	740	0.011
9	(1, 3, 2)	$\lambda = 10, \mu = 10, \alpha_1 = 180, \alpha_2 = 800, \alpha_3 = 460$	1820	1860	0.097
10	(1, 3, 2)	$\lambda = 5, \mu = 5, \alpha_1 = 105, \alpha_2 = 400, \alpha_3 = 270$	990	990	0.046
11	(3, 1, 1, 3)	$\lambda = 5, \mu = 20, \alpha_1 = 1000, \alpha_2 = 420, \alpha_3 = 550, \alpha_4 = 880$	1890	1970	0.095
12	(3, 1, 1, 3)	$\lambda = 10, \mu = 10, \alpha_1 = 550, \alpha_2 = 200, \alpha_3 = 240, \alpha_4 = 480$	1140	1140	0.177
13	(4, 1, 2, 4)	$\lambda = 10, \mu = 5, \alpha_1 = 960, \alpha_2 = 230, \alpha_3 = 600, \alpha_4 = 720$	1330	1380	0.361
14	(4, 1, 2, 4)	$\lambda = 5, \mu = 15, \alpha_1 = 1000, \alpha_2 = 550, \alpha_3 = 800, \alpha_4 = 1200$	2600	2780	0.684
15	(2, 2, 2, 2, 2)	$\lambda = 5, \mu = 5, \alpha_1 = 200, \alpha_2 = 200, \alpha_3 = 400, \alpha_4 = 200, \alpha_5 = 200$	440	460	0.113
16	(2, 2, 2, 2, 2)	$\lambda = 10, \mu = 10, \alpha_1 = 300, \alpha_2 = 300, \alpha_3 = 300, \alpha_4 = 300, \alpha_5 = 300$	860	1000	0.563
17	(3, 3, 3, 3, 3)	$\lambda = 5, \mu = 10, \alpha_1 = 500, \alpha_2 = 500, \alpha_3 = 500, \alpha_4 = 500, \alpha_5 = 500$	1460	1860	30.2826
18	(4, 4, 4, 4, 4)	$\lambda = 5, \mu = 5, \alpha_1 = 800, \alpha_2 = 1300, \alpha_3 = 700, \alpha_4 = 1100, \alpha_5 = 900$	1340	1400	17.3128
19	(6, 2, 3, 6, 1)	$\lambda = 5, \mu = 5, \alpha_1 = 1000, \alpha_2 = 900, \alpha_3 = 500, \alpha_4 = 1100, \alpha_5 = 200$	2880	2970	60.9454

The results reveal that jointly considering tank allocation to the processing steps and robot visiting sequence to the parallel tanks at the processing steps leads to shorter *S*-FOUP cycle time for most cleaning scenarios compared with the traditional method. For Experiments 2, 4, 6, 12, 16, and 17, the bottleneck lies in the robot operations, whereas for the remaining cases, it is governed by one of the processing steps. When the robot becomes the bottleneck of the *S*-FOUP cycle in a wet station, the optimal solutions obtained from the proposed MIP model achieve a substantially shorter cycle time. This is because, when the robot becomes the system bottleneck, the steady-state S-FOUP cycle time of the wet station is primarily determined by the time required for the robot to execute a sequence of operations, including wafer loading/unloading and inter-tank movements. By appropriately

configuring the tank layout and the robot visiting sequence, the total travel distance of the robot within each cycle can be effectively reduced, thereby decreasing the cumulative time spent on movement-related tasks and, consequently, shortening the overall cycle time of the wet station.

This improvement is evident in cases such as Experiment 6, where the proposed MIP model optimizes the tank allocation for wafer flow pattern (4, 2) as (TK_{11}, TK_{12}, TK_{13}, TK_{14}, TK_{21}, TK_{22}) = (T-1, T-2, T-5, T-6, T-3, T-4), and the robot access sequence is $TS_1 = (TK_{14}, TK_{12}, TK_{11}, TK_{13})$, $TS_2 = (TK_{21}, TK_{22})$ instead of (TK_{11}, TK_{12}, TK_{13}, TK_{14}, TK_{21}, TK_{22}) = (T-1, T-2, T-5, T-6, T-3, T-4) with $TS_1 = (TK_{11}, TK_{12}, TK_{13}, TK_{14})$ and $TS_2 = (TK_{21}, TK_{22})$ by the traditional method, which reduces the robot moving distance. Nevertheless, exceptions may exist. For example, the bottleneck in Experiment 12 is the robot that has the heaviest workload, the experimental result achieves the same cycle time as that of the conventional arrangement. When a particular cleaning step emerges as the bottleneck of the S-FOUP cycle in a wet station, under steady-state conditions, the cycle time of a wet station is primarily determined by three components: The processing time of the bottleneck step, the robot loading/unloading time, and the robot's moving time. The processing time and the loading/unloading time are fixed, whereas the robot's traveling time depends solely on the spatial relationship between a given step and its immediate predecessor and successor. In certain cases, the tank layout and the robot visiting sequence do not affect the robot's traveling time. For example, when the first cleaning step constitutes the bottleneck, the location of the preceding Input Buffer is fixed; thus, to minimize the robot's traveling time, the first and second steps are placed as close as possible to the Input Buffer. Similarly, when the last cleaning step becomes the bottleneck, the final and penultimate steps are positioned near the Output Buffer. In both scenarios, the robot's traveling time remains unchanged compared with conventional layouts and, consequently, no reduction in the overall cycle time is achieved. However, when the bottleneck occurs at a certain intermediate cleaning step, whether the cycle time can be reduced depends on the specific wafer flow pattern and the step that is the bottleneck. For instance, in Experiments 1, 3, 7, and 10, the resulting cycle time is identical to that of the classical layout. By contrast, in Experiment 15, the cycle time is shortened against the compared one. In this case, under the wafer flow pattern (2, 2, 2, 2, 2), the proposed MIP model optimizes the tank allocation for wafer flow pattern (4, 2) as (TK_{11}, TK_{12}, TK_{21}, TK_{22}, TK_{31}, TK_{32}, TK_{41}, TK_{42}, TK_{51}, TK_{52}) = (T-5, T-6, T-4, T-10, T-3, T-9, T-2, T-8, T-1, T-7) and the robot visiting sequence as $TS_1 = (TK_{12}, TK_{11})$, $TS_2 = (TK_{22}, TK_{21})$, $TS_3 = (TK_{32}, TK_{31})$, $TS_4 = (TK_{42}, TK_{41})$, and $TS_5 = (TK_{52}, TK_{51})$.

Additionally, in practical semiconductor manufacturing, wet stations are typically configured with approximately 10 tanks to perform wafer cleaning operations. Under such a scale, the MIP model established on the basis of the TPN demonstrates extremely high computational efficiency, as the solution time for all experiments reported in Table 1 is less than one minute. This indicates that the proposed model can rapidly generate effective tank allocation and robot visiting sequences for most actual wafer cleaning scenarios. Ultimately, the practicality of the MIP model proposed in this research has been convincingly validated through multiple experiments, confirming its applicability to handling wet station scheduling problems and its potential to deliver substantial performance improvements.

5. Conclusions

Owing to the linear structure of a wet station, the single-arm robot requires different robot moving time when accessing different tanks in each round, which considerably complicates its scheduling

problem. In this study, we explore the impact of optimizing the allocation of tanks to processing steps and the robot visiting sequence to the parallel tanks at each step. Consequently, it is shown that the tank allocation to the processing steps and the robot visiting sequence to the parallel tanks at each step significantly influence the total traveling distance within a production cycle, thereby affecting the overall cycle time of a wet station. To show this result, we develop an MIP model for wet station scheduling, in which the minimization of cycle time is defined as the objective function by optimizing the tank allocation to processing steps and robot visiting sequence to the parallel tanks at each step. In addition, TPN is employed to specify the time variables of robot actions under the backward strategy and to enforce the corresponding temporal constraints. Experimental validation confirms that, compared with the conventional tank allocation scheme and robot visiting sequence to the parallel tanks, the proposed MIP model can effectively reduce the steady-state cycle time of wet stations across most wafer cleaning scenarios. With the experimental results, it shows that the proposed method reduces the wet station cycle time by 30–400 seconds across scenarios. On average, calculated across the subset of instances where the proposed MIP model outperforms the baseline, this corresponds to a 6.66% reduction compared with the baseline, leading to a 7.14% improvement in production throughput, which is highly significant for large-scale wafer manufacturing. Moreover, the solution time of the proposed MIP model ranges from 1 second to 1 minute, confirming the high computational efficiency of the scheduling approach.

In the subsequent research, we plan to enhance the proposed MIP model by adding additional constraints, such as wafer residency time restrictions. In practical manufacturing, certain cleaning steps in wet stations adopt acidic solutions. Under this processing environment, wafers should be removed from the corresponding tanks in time after processing; otherwise, prolonged immersion could lead to corrosion and compromise product quality. Additionally, in this study, the wet station is assumed to operate under full-load conditions, with every tank utilized in the cleaning process to maximize throughput. However, this setting may result in excessive residency time for wafers in steps with multiple parallel tanks, which is unsuitable when acid-cleaning steps are involved. Therefore, introducing residency time constraints is essential. Under such conditions, not all tanks would necessarily be engaged, creating an inherent conflict between maximizing throughput and ensuring wafer quality. This transforms the wet station scheduling problem into a multi-objective optimization task, requiring future efforts to balance these conflicting objectives.

Acknowledgment

This work was supported in part by the Science and Technology Development Fund (FDCT), Macau SAR (File Nos. 0011/2023/RIA1, 0120/2024/RIA2, and 0004-2024-ITP1).

Data availability

The data will be made available on request.

Use of Generative-Al tools declaration

The authors declare that this paper reports the original research work by the research group and is not generated by Generative-Al tools.

Author contributions

Luetao Li: Formal analysis, Writing-original draft, Methodology; **Naiqi Wu:** Conceptualization, Funding acquisition, Writing-review & editing; **Yan Qiao:** Resources, Software, Investigation; **Siwei Zhang:** Methodology, Supervision, Project administration; **Jie Li**: Investigation, Data curation; **Yang Bai**: Visualization, Validation.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. H. J. Kim, J. H. Lee, T. E. Lee, Scheduling a wet station using a branch and bound algorithm, *Proc. IEEE Int. Conf. Syst. Man Cybern.*, (2012), 2097–2102. https://doi.org/10.1109/ICSMC.2012.6378049
2. N. Q. Wu, C. B. Chu, F. Chu, M. C. Zhou, A Petri net method for schedulability and scheduling problems in single-arm cluster tools with wafer residency time constraints, *IEEE Trans. Semicond. Manuf.*, **21** (2008), 224–237. https://doi.org/10.1109/TSM.2008.2000425
3. C. Jung, T. E. Lee, An efficient mixed integer programming model based on timed Petri nets for diverse complex cluster tool scheduling problems, *IEEE Trans. Semicond. Manuf.*, **25** (2012), 186–199. https://doi.org/10.1109/TSM.2011.2180547
4. V. Borodin, V. Fischer, A. Roussy, C. Yugma, Scheduling semiconductor manufacturing operations in research and development environments, *Proc. SEMI Adv. Semicond. Manuf. Conf. (ASMC)*, (2024), 1–6. https://doi.org/10.1109/ASMC61125.2024.10545483
5. A. C. Huang, S. H. Meng, T. J. Huang, A survey on machine and deep learning in semiconductor industry: methods, opportunities, and challenges, *Cluster Comput.*, **26** (2023), 3437–3472. https://doi.org/10.1007/s10586-023-04115-6
6. S. Bhushan, I. A. Karimi, An MILP approach to automated wet-etch station scheduling, *Ind. Eng. Chem. Res.*, **42** (2003), 1391–1399. https://doi.org/10.1021/ie020296c
7. J. H. Pang, H. M. Zhou, Y. C. Tsai, F. D. Chou, A scatter simulated annealing algorithm for the bi-objective scheduling problem for the wet station of semiconductor manufacturing, *Comput. Ind. Eng.*, **123** (2018), 54–66. https://doi.org/10.1016/j.cie.2018.06.017
8. J. H. Han, S. H. Jeong, G. Hwang, J. Y. Lee, Machine learning-based dispatching for a wet clean station in semiconductor manufacturing, *J. Manuf. Syst.*, **77** (2024), 341–355. https://doi.org/10.1016/j.jmsy.2024.09.018
9. T. L. Perkinson, P. K. Maclarty, R. S. Gyurcsik, R. K. Cavin III, Single-wafer cluster tool performance: An analysis of throughput, *IEEE Trans. Semicond. Manuf.*, **7** (1994), 369–373. https://doi.org/10.1109/66.311340
10. T. L. Perkinson, R. S. Gyurcsik, P. K. Maclarty, Single-wafer cluster tool performance: An analysis of effects of redundant chambers and revisitation sequences on throughput, *IEEE Trans. Semicond. Manuf.*, **9** (1996), 384–400. https://doi.org/10.1109/66.536110

11. S. Venkatesh, R. Davenport, P. Foxhoven, J. Nulman, A steady state throughput analysis of cluster tools: Dual-blade versus single-blade robots, *IEEE Trans. Semicond. Manuf.*, **10** (1997), 418–424. https://doi.org/10.1109/66.641483
12. C. Jung, H. J. Kim, T. E. Lee, A branch and bound algorithm for cyclic scheduling of timed Petri nets, *IEEE Trans. Autom. Sci. Eng.*, **12** (2015), 309–323. https://doi.org/10.1109/TASE.2013.2285221
13. D. Suerich, T. McIlroy, Artificial intelligence for real-time cluster tool scheduling: EO equipment optimization, *Proc. SEMI Adv. Semicond. Manuf. Conf. (ASMC)*, (2022). https://doi.org/10.1109/ASMC54647.2022.9792523
14. J. Ahn, H. J. Kim, A novel mixed integer programming model with precedence relation-based decision variables for non-cyclic scheduling of cluster tools, *IEEE Trans. Autom. Sci. Eng.*, **22** (2025), 2893–2908. https://doi.org/10.1109/TASE.2024.3386562
15. T. E. Lee, H. Y. Lee, Y. H. Shin, Workload balancing and scheduling of a single-armed cluster tools, *Proc. Asia-Pacific Ind. Eng. Manag. Syst. Conf.*, (2004), 1–6.
16. J. H. Lee, H. J. Kim, Backward sequence analysis for single-armed cluster tools, *Proc. Int. Conf. Autom. Plan. Sched.*, **29** (2019), 269–272. https://doi.org/10.1609/icaps.v29i1.3395
17. Y. H. Shin, T. E. Lee, J. H. Kim, H. Y. Lee, Modeling and implementing a real-time scheduler for dual-armed cluster tools, *Comput. Ind.*, **45** (2001), 13–27. https://doi.org/10.1016/S0166-3615(01)00078-1
18. N. Q. Wu, M. C. Zhou, Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation, *IEEE Trans. Autom. Sci. Eng.*, **9** (2012), 203–209. https://doi.org/10.1109/TASE.2011.2160452
19. J. H. Kim, Stable schedule for a single-armed cluster tool with time constraints, *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, (2008), 97–102. https://doi.org/10.1109/COASE.2008.4626464
20. Q. H. Zhu, Y. Qiao, N. Q. Wu, Y. Hou, Post-processing time-aware optimal scheduling of single robotic cluster tools, *IEEE/CAA J. Autom. Sin.*, **7** (2020), 597–605. https://doi.org/10.1109/JAS.2020.1003069
21. W. Q. Xiong, C. R. Pan, Y. Qiao, N. Q. Wu, M. X. Chen, P. H. Hsieh, Reducing wafer delay time by robot idle time regulation for single-arm cluster tools, *IEEE Trans. Autom. Sci. Eng.*, **18** (2021), 1653–1667. https://doi.org/10.1109/TASE.2020.3014078
22. J. H. Kim, T. E. Lee, H. Y. Lee, D. B. Park, Scheduling analysis of timed-constrained dual-armed cluster tools, *IEEE Trans. Semicond. Manuf.*, **16** (2003), 521–534. https://doi.org/10.1109/TSM.2003.815203
23. S. Rostami, B. Hamidzadeh, Optimal scheduling techniques for cluster tools with process-module and transport-module residency constraints, *IEEE Trans. Semicond. Manuf.*, **15** (2002), 341–349. https://doi.org/10.1109/TSM.2002.801379
24. Y. Lim, T. S. Yu, T. E. Lee, A new class of sequences without interferences for cluster tools with tight wafer delay constraints, *IEEE Trans. Autom. Sci. Eng.*, **16** (2019), 392–405. https://doi.org/10.1109/TASE.2018.2815157
25. D. H. Roh, T. G. Lee, T. E. Lee, K-cyclic schedules and the worst-case wafer delay in a dual-armed cluster tool, *IEEE Trans. Semicond. Manuf.*, **32** (2019), 236–249. https://doi.org/10.1109/TSM.2019.2910399
26. C. Kim, T. E. Lee, Feedback control of cluster tools for regulating wafer delays, *IEEE Trans. Autom. Sci. Eng.*, **13** (2016), 1189–1199. https://doi.org/10.1109/TASE.2015.2404921

27. R. Jacob, S. Amari, Output feedback control of discrete processes under time constraint: application to cluster tools, *Int. J. Comput. Integr. Manuf.*, **30** (2017), 880–894. https://doi.org/10.1080/0951192X.2016.1224391
28. D. H. Roh, T. E. Lee, C. Martinez, Kanban feedback control for wafer delay regulation of cluster tools, *IEEE Trans. Autom. Sci. Eng.*, **22** (2025), 6822–6838. https://doi.org/10.1109/TASE.2024.3455253
29. T. S. Yu, H. J. Kim, T. E. Lee, Scheduling single-armed cluster tools with chamber cleaning operations, *IEEE Trans. Autom. Sci. Eng.*, **15** (2018), 705–716. https://doi.org/10.1109/TASE.2017.2682271
30. T. S. Yu, T. E. Lee, Scheduling dual-armed cluster tools with chamber cleaning operations, *IEEE Trans. Autom. Sci. Eng.*, **16** (2019), 218–228. https://doi.org/10.1109/TASE.2017.2764105
31. L. Zhen, B. Zhang, F. Yang, M. Zhou, Efficient scheduling of residency-time-constrained dual-armed cluster tools with condition-based chamber cleaning operations, *IEEE Trans. Autom. Sci. Eng.*, **22** (2025), 5000–5011. https://doi.org/10.1109/TASE.2024.3414843
32. M. X. Liu, B. H. Zhou, Modelling and scheduling analysis of multi-cluster tools with residency constraints based on time constraint sets, *Int. J. Prod. Res.*, **51** (2013), 4835–4852. https://doi.org/10.1080/00207543.2013.774490
33. T. Bao, H. Wang, Cyclic scheduling of multi-cluster tools based on mixed integer programming, *IEEE Trans. Semicond. Manuf.*, **30** (2017), 515–525. https://doi.org/10.1109/TSM.2017.2733559
34. X. Li, C. Lu, Scheduling dual-arm multi-cluster tools with residency time constraints beyond swap-based strategies and module-bound regions, *IEEE Robot. Autom. Lett.*, **9** (2024), 10177–10184. https://doi.org/10.1109/LRA.2024.3469823
35. Q. H. Zhu, G. X. Zhang, Y. Hou, N. Q. Wu, Scheduling a dual-arm robotic two-cluster tool in the event of a process module failure, *Expert Syst. Appl.*, **297** (2026), https://doi.org/10.1016/j.eswa.2025.129415
36. S. G. Ko, T. S. Yu, T. E. Lee, Scheduling dual-armed cluster tools for concurrent processing of multiple wafer types with identical job flows, *IEEE Trans. Autom. Sci. Eng.*, **16** (2019), 1058–1070. https://doi.org/10.1109/TASE.2018.2868004
37. J. Wang, T. Leng, C. Liu, M. Zhou, S. Zhao, Scheduling of single-arm cluster tools handling multiple wafer types based on double-layer configuration of processing modules, *IEEE Trans. Autom. Sci. Eng.*, **21** (2024), 6833–6844. https://doi.org/10.1109/TASE.2023.3332551
38. B. Li, J. Zhong, W. L. Liu, H. Jin, End-to-end scheduling of single-arm cluster tools with multiple wafer types, *Complex Syst. Model. Simul.*, (2025). https://doi.org/10.23919/CSMS.2025.0012
39. T. Murata, Petri nets: Properties, analysis and applications, *Proc. IEEE*, **77** (1989), 541–580. https://doi.org/10.1109/5.24143
40. W. M. Zuberek, Timed Petri nets in modeling and analysis of cluster tools, *IEEE Trans. Robot. Autom.*, **17** (2001), 562–575. https://doi.org/10.1109/70.964658
41. W. M. Zuberek, Cluster tools with chamber revisiting: Modeling and analysis using timed Petri nets, *IEEE Trans. Semicond. Manuf.*, **17** (2004), 333–344. https://doi.org/10.1109/TSM.2004.831524
42. M. Jeng, Comments on timed Petri nets in modeling and analysis of cluster tools, *IEEE Trans. Autom. Sci. Eng.*, **2** (2005), 92–93. https://doi.org/10.1109/TASE.2004.840069

43. N. Q. Wu, Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model, *IEEE Trans. Syst. Man Cybern. C*, **29** (1999), 192–204. https://doi.org/10.1109/5326.760564
44. N. Q. Wu, M. C. Zhou, System Modeling and Control with Resource-Oriented Petri Nets, CRC Press, (2010). https://doi.org/10.1201/9781439808856
45. Y. Qiao, N. Q. Wu, M. C. Zhou, Scheduling of dual-arm cluster tools with wafer revisiting and residency time constraints, *IEEE Trans. Ind. Inform.*, **10** (2014), 286–300. https://doi.org/10.1109/TII.2013.2272702
46. N. Q. Wu, 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. Sci. Eng.*, **9** (2012), 446–454. https://doi.org/10.1109/TASE.2011.2178023
47. Y. Qiao, N. Q. Wu, M. C. Zhou, Real-time scheduling of single-arm cluster tools subject to residency time constraints and bounded activity time variation, *IEEE Trans. Autom. Sci. Eng.*, **9** (2012), 564–577. https://doi.org/10.1109/TASE.2012.2192476
48. Q. H. Zhu, M. C. Zhou, Y. Qiao, N. Q. Wu, Scheduling transient processes for time-constrained single-arm robotic multi-cluster tools, *IEEE Trans. Semicond. Manuf.*, **30** (2017), 261–269. https://doi.org/10.1109/TSM.2017.2721970
49. F. Yang, Y. Qiao, K. Z. Gao, N. Q. Wu, Y. T. Zhu, I. W. Simon, Efficient approach to scheduling of transient processes for time-constrained single-arm cluster tools with parallel chambers, *IEEE Trans. Syst. Man Cybern. Syst.*, **50** (2020), 3646–3657. https://doi.org/10.1109/TSMC.2018.2852724
50. L. Chen, Y. He, B. Zhang, C. Xia, Y. Qiu, Z. Yu, Petri net-based information enhancement for quality monitoring in the cigarette manufacturing process, *Measurement*, **255** (2025), 115000. https://doi.org/10.1016/j.measurement.2025.117999
51. J. Luo, S. Yi, Z. Lin, H. Zhang, J. Zhou, Petri-net-based deep reinforcement learning for real-time scheduling of automated manufacturing systems, *J. Manuf. Syst.*, **74** (2024), 995–1008. https://doi.org/10.1016/j.jmsy.2024.05.006
52. M. R. Hajiabadi, R. H. Amlashi, A. A. R. Hosseinabadi, S. Mirkamali, G. W. Weber, Two-machine flow shop task scheduling using a hybrid gravitational search algorithm called SAGSA, *J. Ind. Manag. Optim.*, **21** (2025), 4815–4840. https://doi.org/10.3934/jimo.2025075
53. Y. H. Chen, H. W. Wang, C. H. Tsai, C. H. Wu, A genetic algorithm for tool-constrained scheduling with dynamic expedited orders in smart manufacturing, *J. Ind. Manag. Optim.*, **22** (2026), 182–213. https://doi.org/10.3934/jimo.2026007
54. F. Qiao, Y. M. Ma, L. Li, H. X. Yu, A Petri net and extended genetic algorithm combined scheduling method for wafer fabrication, *IEEE Trans. Autom. Sci. Eng.*, **10** (2013), 197–204. https://doi.org/10.1109/TASE.2012.2204049
55. C. Zhao, H. Zhang, C. Fan, Z. Gao, X. Zhu, Maintenance scheduling of semiconductor production equipment based on particle swarm optimization, *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci.*, (2020), 436–439. https://doi.org/10.1109/ICSESS49938.2020.9237744
56. C. D. Geiger, K. G. Kempf, R. Uzsoy, A tabu search approach to scheduling an automated wet etch station, *J. Manuf. Syst.*, **16** (1997), 102–116. https://doi.org/10.1016/S0278-6125(97)85674-9
57. S. Bhushan, I. A. Karimi, Heuristic algorithms for scheduling an automated wet-etch station, *Comput. Chem. Eng.*, **28** (2004), 363–379. https://doi.org/10.1016/S0098-1354(03)00192-3

58. T. E. Lee, H. Y. Lee, S. J. Lee, Scheduling a wet station for wafer cleaning with multiple job flows and multiple wafer-handling robots, *Int. J. Prod. Res.*, **45** (2007), 487–507. https://doi.org/10.1080/00207540600792531
59. A. M. Aguirre, C. A. Méndez, P. M. Castro, A novel optimization method to automated wet-etch station scheduling in semiconductor manufacturing systems, *Comput. Chem. Eng.*, **35** (2011), 2960–2972. https://doi.org/10.1016/j.compchemeng.2011.02.014
60. J. M. Novas, G. P. Henning, A comprehensive constraint programming approach for the rolling horizon-based scheduling of automated wet-etch stations, *Comput. Chem. Eng.*, **42** (2012), 189–205. https://doi.org/10.1016/j.compchemeng.2012.01.005

AIMS Press