Modeling and Optimal Scheduling of Wafer-residency-time Constrained Cluster Tools via Petri Nets and Linear Programming

FaJun Yang, NaiQi Wu, *Senior Member*, *IEEE*, Yan Qiao, Rong Su, *Senior Member*, *IEEE*, MengChu Zhou, *Fellow*, *IEEE*

***Abstract*: Scheduling a cluster tool with wafer residency time constraint is of great challenge and importance in wafer manufacturing. With a backward strategy, the scheduling problem of single-robot-arm cluster tools with such a constraint is well-studied in the literature. It is much more challenging to schedule a more general case whose optimal scheduling strategy is not limited to the backward strategy. This work uses a timed Petri net to model the dynamic behavior of the system and presents a method to find the optimal scheduling strategy for the system. Based on the Petri net model and obtained strategy, it reveals that the key issue to schedule such a tool is to determine when and how long the robot should wait for. Based on this, this work establishes for the first time the necessary and sufficient conditions regarding the existence of an optimal and feasible one-wafer cyclic schedule for single-robot-arm cluster tools. It then establishes a computationally efficient linear program to find it if existing, and finally gives industrial examples to show the application and power of the proposed method.**

***Index Terms*—Wafer manufacturing, cluster tools, Petri net (PN), scheduling, parallel modules.**

I. Introduction

As a kind of robotic manufacturing systems that implement single-wafer processing technology, cluster tools are proved to provide a flexible, reconfigurable, and efficient environment [Bader, *et al*., 1990; and Burggraaf, 1995], resulting in greater productivity [Newboe, 1990], shorter cycle time [McNab, 1990; Newboe, 1990; and Singer, 1995], higher utilization of space [Burggraaf, 1995; and Singer, 1995], and lower capital expenses [Singer, 1995]. This answers why they are increasingly adopted by semiconductor manufacturers worldwide. A cluster tool usually consists of a number of wafer process modules (PMs), two loadlocks for wafer loading/unloading, and a robot as shown in Fig. 1. The robot is equipped with a single arm or dual arms, correspondingly, leading to a single- or dual-robot-arm cluster tool.

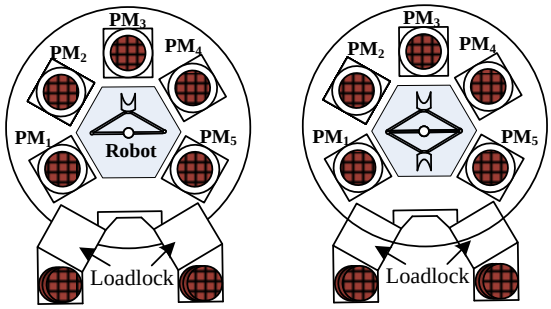


Fig. 1. (a) single-robot-arm tool and (b) dual-robot-arm tool.

Through the loadlocks, raw wafers are dropped into a cluster tool in a cassette-by-cassette way and a cassette holds a lot of wafers that own a same predefined routing sequence, called a recipe [Kim, *et al*., 2003; and Lee and Park, *et al*., 2005]. The robot is instructed to unload a wafer from the cassette and transport it from one PM to another for processing. At each step, a wafer should stay in a PM for some time to be processed and return to the cassette from which it came at last [Wu and Zhou, 2010b]. If all the wafers in a cassette complete their processing, the cassette is unloaded from the loadlock such that another one can be loaded. With two loadlocks, a cluster tool can run continuously without interruption.

For some wafer manufacturing processes, such as low pressure chemical vapor deposition (LPCVD), a processed wafer needs to be removed from the PM within a limited time. Otherwise its surface would suffer from severe quality problems due to residual gases and high temperature in a PM [Rostami *et al.*, 2001; Kim *et al.*, 2003; and Lee *et al.*, 2005]. Such a constraint is referred to as wafer residency time constraint in [Rostami *et al.*, 2001], which greatly complicates

This work was supported in part by FDCT of Macau under Grants 065/2013/A2 and 066/2013/A2. And it was also conducted with funding support from Delta Electronics Inc and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

F. J. Yang and R. Su are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore, Singapore (e-mail: fjyang@ntu.edu.sg, RSu@ntu.edu.sg).

N. Q. Wu is with the Institute of Systems Engineering, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau, and also the Department of Industrial Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: nqwu@must.edu.mo).

Y. Qiao is with the Institute of Systems Engineering, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau (e-mail: dr.yqiao@gmail.com)

M. C. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982 USA (e-mail: zhou@njit.edu).

the scheduling problem of cluster tools.

[Rostami, *et al*. 2001] presents heuristic algorithms to find an optimal cyclic schedule for dual-robot-arm cluster tools with wafer residency time constraint. However, the existence of a feasible schedule is not investigated. In [Kim, *et al*., 2003], schedulability is analyzed for dual-robot-arm cluster tools with wafer residency time constraint and a linear programming-based method is developed to find a feasible and optimal schedule if schedulable. However, the obtained conditions are just sufficient but not necessary. [Lee, *et al*. 2005] establishes necessary and sufficient conditions under which the system is schedulable and develops an algorithm to obtain an optimal schedule by using an integer programming formulation. Its drawback is its complex computation.

The scheduling problem of both single- and dual-robot-arm cluster tools with wafer residency time constraint is further investigated in [Wu *et al.*, 2008; and Wu and Zhou, 2010b]. A type of generic PN models is invented to analyze the relationship between the wafer residency time in PMs and the cycle time of the system, and necessary and sufficient conditions under which a system is schedulable are proposed. Furthermore, an optimal and feasible one-wafer cyclic schedule, which means that during each cycle time, only one wafer is loaded into and unloaded from the loadlocks, is obtained by analytical expressions. Clearly, such methods represent the most efficient ones.

In recent years, [Qiao *et al*., 2014 and 2015] tackled the scheduling problem of dual-robot-arm cluster tools with both wafer residency time constraint and revisiting processing. The studies [Wu and Zhou, 2012 and Qiao *et al*., 2012] analyzed the effect of activity time variation on wafer residency time delay in a PM for dual and single-robot-arm cluster tools. In [Qiao *et al*., 2012], the study is conducted under the assumption that a backward strategy is optimal for single-robot-arm cluster tools. This assumption does not hold unfortunately for some cases as shown in [Dawande *et al.*, 2002].

In [Dawande *et al.*, 2002, and Chan *et al.*, 2011], the scheduling strategy is not limited to be backward. To find an optimal one-unit (or one-wafer) cyclic schedule for robot cells with a single gripper robot, a polynomial time algorithm is proposed in [Dawande *et al.*, 2002]. As cluster tools are a type of robotic cells, such an algorithm can clearly be applied to single-robot-arm cluster tools. Although Dawande *et al.* [2002] can determine the robot task sequence to obtain an optimal cyclic schedule for a single-robot-arm cluster tool without wafer residency time constraint, the obtained schedule may not be feasible if such a constraint is imposed and further research is, hence, necessary to address this new issue. This work intends to address it for the first time to our knowledge. It considers the general situation that multiple PMs can be configured for a processing step, which relaxes the restriction made in [Dawande *et al.*, 2002] where only one PM is configured for a step.

To solve the above problem, we adopt a Petri net model to describe the dynamic behavior of the system. With the model, the method in [Dawande *et al.*, 2002] is used to determine the optimal scheduling strategy. Based on an obtained strategy, we analyze the relationship between the wafer residency time in PMs and the cycle time of the system such that a schedule is parameterized by robot waiting time. Then, we establish necessary and sufficient conditions under which an optimal and feasible schedule exists. With these conditions, such a schedule is efficiently found by solving a linear program. In this way, the open problem is solved.

To compare our work with others, we conclude that the work in [Rostami, *et al*. 2001, Kim, *et al*., 2003, Wu and Zhou, 2010b, 2012, Qiao *et al*., 2014 and 2015] only aims to the dual-robot-arm cluster tools. [Wu *et al.*, 2008, Qiao *et al*., 2012, Dawande *et al.*, 2002, and Chan *et al.*, 2011] focus on single-robot-arm cluster tools. However, in [Wu *et al.*, 2008, Qiao *et al*., 2012], it is assumed that a backward strategy is always optimal. In [Dawande *et al.*, 2002, and Chan *et al.*, 2011] however, no wafer residency time constraint is considered. For [Lee, *et al*. 2005], though wafer residency time constraint is taken into account for both single- and dual-robot-arm cluster tools, its drawback is its complex computation. In summary, this work makes the following two primary contributions:

1) Derives for the first time the necessary and sufficient conditions regarding the existence of an optimal and feasible one-wafer cyclic schedule for a wafer-residency-time constrained single-robot-arm cluster tool whose backward scheduling strategy may not be optimal; and

2) Establishes a computationally efficient linear program to find such a cyclic schedule if existing.

In the next section, we develop the PN model for the system. Section III discusses how to determine the optimal scheduling strategy and the relationship between the wafer sojourn time in PMs and cycle time of the system. Thereafter, Section IV presents the necessary and sufficient conditions under which an optimal and feasible one-wafer cyclic schedule exists and the linear program for finding it. Examples are given to show the applications of the proposed method in Section V. Finally, Section VI concludes this paper.

II. System Modeling

Let $\mathbf{N}_n = \{1, 2, \ldots, n\}$ and $\mathbf{\Omega}_n = \mathbf{N}_n \cup \{0\}$. To balance the workloads among the steps, parallel PMs can be configured for some steps. Thus, a general wafer flow pattern is denoted as $(m_1, m_2, \ldots, m_n)$, where n is the number of processing steps and m_i, $i \in \mathbf{N}_n$, denotes the number of parallel PMs at Step i [Kim *et al.*, 2003].

A. Finite Capacity Petri Net

As PNs can deal with concurrent activities, they are widely used as an effective tool for modeling, analysis, and control of manufacturing systems [Wu *et al.*, 2008, and 2011, 2013; and Kim *et al.*, 2003; Hu *et al.*, 2015a and b]. To operate a cluster tool, we need to allocate the finite resources to the PMs. Since PNs are powerful in modeling the behavior of resource allocation, a finite capacity PN is an ideal choice to model a cluster tool. In this work, we model a single-robot-arm cluster

tool by extending the PN developed in [Wu, 1999, 2008; Wu and Zhou, 2001 and 2010a] and its concept is based on [Zhou and Venkatesh, 1998; and Wu and Zhou 2009].

The PN model here is defined as PN = (P, T, I, O, M, K), where P and T are finite sets of places and transitions; I/O is an input/output function; M is a marking gives the number of tokens in places with M_0 presenting the initial one; and K is a capacity function, where $K(p)$ denotes the maximum number of tokens that place p can hold at a time. The preset of transition t is the set of all input places to t, i.e., ${}^\bullet t = \{p: p \in P \text{ and } I(p, t) > 0\}$, while its postset is the set of all output places from t, i.e., $t^\bullet = \{p: p \in P \text{ and } O(p, t) > 0\}$. Similarly, we have p's preset ${}^\bullet p = \{t \in T: O(p, t) > 0\}$ and its postset $p^\bullet = \{t \in T: I(p, t) > 0\}$. For the transition enabling and firing rules, readers can refer to [Wu, 1999; Wu and Zhou, 2001 and 2010a].

B. *Petri Net for the System*

This work extends the well-developed PN model in [Wu *et al.*, 2008] to model a single-robot-arm cluster tool. For convenience, a token and a wafer are often used without difference in introducing the PN model.

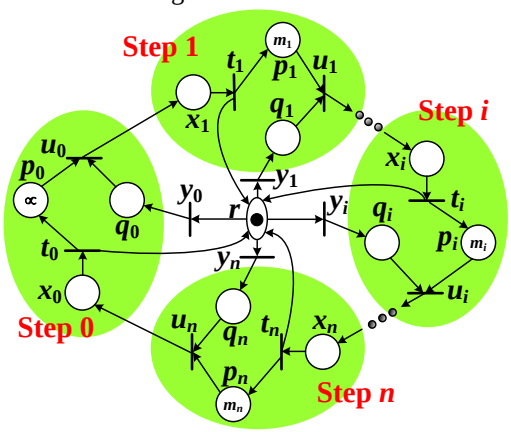


Fig. 2. PN model for system with n + 1 steps.

Assume that the loadlocks are numbered as Step 0. The n processing steps are numbered as Steps 1 to n. For concise presentation, we assume that Step n + 1 is identical to Step 0, or n + 1 is treated as 0. Then, with n processing steps, there are n + 1 steps for the system. PMs at Step i, $i \in \mathbf{N}_n$, are modeled by timed place p_i with $K(p_i) = m_i$. To represent that there are always wafer to be processed in the loadlocks, we have $K(p_0) = \infty$. Timed place q_i with $K(q_i) = 1$ models the robot's waiting before removing a wafer from p_i, $i \in \mathbf{\Omega}_n$. Timed place x_i, $i \in \mathbf{\Omega}_n$, with $K(x_i) = 1$ models that the robot moves from Steps i-1 to i with a wafer carried. Pictorially, these places are denoted by a circle. Timed transitions t_i and u_i model that the robot drops a wafer into p_i and removes a wafer from p_i, respectively. Timed transition y_i models that the robot moves from Steps j to i, j, $i \in \mathbf{\Omega}_n$, without carrying a wafer. Pictorially, transitions are denoted by a bar. Place r with $K(r) = 1$ models the robot and it is denoted by an ellipse. If there is a token in r, the robot is available and can be used to unload a wafer. An arc (t_i, r), $i \in \mathbf{\Omega}_n$, is added to represent that, by firing t_i (loading a wafer into p_i), the robot arm is released. At last, by adding arcs (x_i, t_i), (t_i, p_i), (p_i, u_i), (q_i, u_i), and (u_i, x_{i+1}), $i \in \mathbf{\Omega}_n$, the PN model for the system is obtained as shown in Fig. 2.

In the PN model in Fig. 2, when M(r) = 1, any y_i, $i \in \mathbf{\Omega}_n$, is enabled and can fire. In [Wu *et al.*, 2008], to model the backward strategy, a simple control policy is posed to control the firing of y_i's. This work studies the scheduling problem of a more general case whose the optimal schedule strategy is not limited to be backward, hence, we need to find an optimal strategy that presents the robot task sequence. Note that such a strategy is dependent on the wafer processing time at the steps and robot task time for a given scenario. For the problem addressed in this work, when a strategy is determined, the robot task sequence is determined as well, which is acted as a control policy to control the firing of y_i's such that the system is correctly modeled.

To describe a scheduling strategy we define a basic activity A_i for Step i with $A_i = \langle y_i \to u_i \to x_{i+1} \to t_{i+1} \rangle$, $i \in \mathbf{\Omega}_n$. Note that if t_i is followed by y_i, then, y_i represents that the robot moves from Step i to itself, or it takes no action if $m_i = 1$. Otherwise, if $m_i > 1$, it also needs to take a time for moving from one module to another within Step i. Then, without taking the robot waiting into account, $\eta = (A_{i_0} A_{i_1} \ldots A_{i_n})$ can be used to describe a scheduling strategy for the system, where i_0, i_1, …, and i_n denote a permutation of steps {0, 1, 2, …, n} [Sechi *et al.*, 1992, Crama *et al.*, 1997, Dawande *et al.*, 2002, and Chan *et al.*, 2011]. In this work, it is assumed that η begins with unloading a wafer from Step 0. Thus, we always have $A_{i_0} = A_0$, unless otherwise specified.

Note that a backward strategy forms a η, i.e., a backward strategy will be applied when it is optimal. For easy presentation, we need the following definitions that are cited from [Dawande *et al.*, 2002, and Chan *et al.*, 2011].

Definition 2.1 [Dawande *et al.*, 2002]: A one-wafer cycle is a sequence of activities with each basic activity A_i, $i \in \mathbf{\Omega}_n$, appearing just once and during which one wafer is dropped into and removed from the loadlocks. The cycle time is the minimal time needed to complete such a one-wafer cycle.

Definition 2.2 [Dawande *et al.*, 2002, and Chan *et al.*, 2011]: In a scheduling startegy η, an activity chain is a sequence of activities whose indexes appear with a consecutively increasing order, but not necessary adjacently.

Take $\eta = (A_0A_4A_5A_1A_2A_3)$ as an example. By Definition 2.2, A_0 and $A_1A_2A_3$ form an activity chain although A_0 and $A_1A_2A_3$ are separated by A_4A_5. Also, A_4A_5 forms an activity chain. Hence, there are two activity chains in $\eta = (A_0A_4A_5A_1A_2A_3)$.

With a scheduling strategy being determined and the above obtained PN structure, based on Definitions 2.1 and 2.2, we need to set the initial marking M_0 of the model next. Also, we need to control the PN model such that the robot scheduling strategy can be exactly realized with no deadlock. This can be done by introducing a control policy. At last, for the PN model, we associate activity time with both transitions and places to describe the temporal aspect of the system. In such a way, we find that to schedule the system is to determine when and how long the robot should wait for.

To set the initial marking M_0 of the model, from [Chan *et al.* 2011], in each activity chain, during the whole cycle, only one wafer should be being processed, due to their assumption that each step consists of only one PM. For the problem addressed in this work, we have m_i ($m_i \geq 1$) PMs for Step i, $i \in \mathbf{N}_n$. Hence, we first assume that, at Step i, $i \in \mathbf{N}_n$, m_i - 1 wafers are being processed by m_i - 1 PMs, respectively. Then, for the remaining n PMs (not including the loadlocks), according to [Chan *et al.* 2011], in each activity chain, only one wafer should be being processed during the whole cycle. By doing so, if an optimal scheduling strategy is applied and schedulable, the maximum productivity can be obtained. Thus, given a scheduling strategy η, by putting a type of tokens V_0 representing a virtual wafer into the PMs, we can set the initial marking M_0 as follows.

Let $\mathbf{D}$ be the set of step indexes of an activity chain in η. Let k = Min{$\mathbf{D}$}. Then, set the initial marking as follows. For each activity chain in η, find $\mathbf{D}$ and set $M_0(p_k) = m_k$, $k \neq 0$, $M_0(p_j) = m_j$ - 1, $j \in \mathbf{D}\backslash\{0, k\}$; $M_0(r) = 0$; $M_0(x_i) = 0$, $i \in \mathbf{\Omega}_n$; $M_0(q_j) = 0$, $j \in \mathbf{N}_n$; and $M_0(q_0) = 1$, indicating that the robot is waiting at Step 0 for unloading a wafer from the loadlocks.

For example, given $\eta = (A_0A_5A_3A_4A_1A_2)$, we have three activity chains: 1) $A_0A_1A_2$; 2) A_5; and 3) A_3A_4. For Chain 1, $\mathbf{D} = \{0, 1, 2\}$, the minimal one is 0. Thus, we set $M_0(p_1) = m_1$ - 1, and $M_0(p_2) = m_2$ - 1. For Chain 2, $\mathbf{D} = \{5\}$ and we set $M_0(p_5) = m_5$. For Chain 3, $\mathbf{D} = \{3, 4\}$ and Min{3, 4} = 3. Thus, we set $M_0(p_3) = m_3$, and $M_0(p_4) = m_4$ - 1. Then, set $M_0(r) = 0$, $M_0(x_i) = 0$, $i \in \mathbf{\Omega}_n$. Finally, set $M_0(q_j) = 0$, $j \in \mathbf{N}_n$, and $M_0(q_0) = 1$.

At M_0, as $M_0(r) = 0$; $M_0(x_i) = 0$, $i \in \mathbf{\Omega}_n$; $M_0(q_j) = 0$, $j \in \mathbf{N}_n$, and $M_0(q_0) = 1$; the only enabled transition is u_0. After u_0 fires, the robot performs task sequence: $\langle x_1 \rightarrow t_1 \rangle$ to load a wafer into Step 1 such that M_1 is reached, and at M_1 we have $M_1(p_1) = m_1$. Thereafter, the robot may execute the following task sequence: $\langle y_0 \rightarrow q_0 \rightarrow u_0 \rightarrow x_1 \rangle$ to load a wafer into Step 1 again. As $M_1(p_1) = m_1$, from the transition enabling and firing rules in [Wu, 1999; Wu and Zhou, 2001], a deadlock occurs. To avoid such a deadlock, we have the following control policy.

Definition 2.3: For the PN of the system, given a strategy $\eta = (A_{i_0} A_{i_1} \ldots A_{i_n})$, at marking M, y_{i_j}, $j \in \mathbf{N}_n$, is said to be control-enabled if $t_{(i_{j-1})+1}$ has just executed; y_{i_0} (or y_0) is said to be control-enabled if $t_{(i_n)+1}$ has just executed.

By Definition 2.3, at M_0, after u_0 fires, the robot performs a task sequence $\langle x_1 \rightarrow t_1 \rightarrow y_{i_1} \rightarrow u_{i_1} \rightarrow x_{(i_1)+1} \rightarrow t_{(i_1)+1} \rightarrow y_{i_2} \rightarrow u_{i_2} \rightarrow x_{(i_2)+1} \rightarrow t_{(i_2)+1} \rightarrow y_{i_3} \rightarrow u_{i_3} \rightarrow \ldots u_{i_n} \rightarrow x_{(i_n)+1} \rightarrow t_{(i_n)+1} \rightarrow y_{i_0} (y_0) \rightarrow u_0 \rightarrow x_1 \rangle$ such that a cycle is completed and there is no deadlock. In this way, the PN model is made deadlock-free.

C. Modeling Activity Time

In the PN model, both transitions and places are associated with time. If transition t is associated with time ς, it means that firing t lasts for ς time units. If place p is associated with time ς, it implies that a token in p has to stay there for at least ς time units before it can enable its output transition [Wu *et al.*, 2008].

As done in [Kim *et al.*, 2003; Geismar *et al.*, 2004; and Lee and Lee, 2006], we assume that: 1) the time taken for the robot to unload/load a wafer from/into a PM and loadlocks is identical and denoted by λ, and 2) the time taken for the robot to move from Steps j to i, $j \neq i$, is same no matter whether it carries a wafer or not and denoted by μ. The time required to process a wafer at Step i is α_i, $i \in \mathbf{N}_n$, and $\alpha_0 = 0$. ω_i, $i \in \mathbf{\Omega}_n$, denotes the robot waiting time in q_i, and τ_i, $i \in \mathbf{N}_n$, the wafer sojourn time at Step i. After being processed, a wafer needs to be removed from a PM at Step i within a limited time δ_i, $i \in \mathbf{N}_n$, we have $\tau_i \in [\alpha_i, \alpha_i + \delta_i]$. In the loadlocks, there are no wafer processing and thus there is no wafer residency time constraint.

Table I. Time duration associated with transitions and places.

Symbol	Transition or place	Action	Time duration
λ	$t_i \in T$	Robot loads a wafer into Step i, $i \in \mathbf{\Omega}_n$	λ
	$u_i \in T$	Robot unloads a wafer from Step i, $i \in \mathbf{\Omega}_n$	
μ	$y_i \in T$	Robot moves from Steps j to i, $j \neq i$ and j, $i \in \mathbf{\Omega}_n$, without holding a wafer	μ
	$y_i \in T$	Robot moves from one PM to another within Step i when $m_i > 1$, $i \in \mathbf{N}_n$, without holding a wafer	μ
	$y_i \in T$	Robot moves from Step i to itself with $m_i = 1$, $i \in \mathbf{\Omega}_n$, without holding a wafer	0
	$x_i \in P$	Robot moves from Steps i-1 to i, $i \in \mathbf{N}_n$, with a wafer being held	μ
	$x_0 \in P$	Robot moves from Step n to 0, with a wafer being held	
τ_i	$p_i \in P$	A wafer being processed and waiting in p_i, $i \in \mathbf{N}_n$	$[\alpha_i, \alpha_i + \delta_i]$
ω_i	$q_i \in P$	Robot waits before unloading a wafer from Step i, $i \in \mathbf{\Omega}_n$	≥ 0

If activity sequence $A_{i-1}A_i/A_nA_0$ is an item in η, after loading a wafer into p_i/p_0, the robot needs to wait there for at least α_i time units if $m_i = 1$, $i \in \mathbf{N}_n$, and then unloads a wafer from p_i/p_0. This implies that y_i/y_0 takes no time. However, if $m_i > 1$, after loading a wafer into Step i, the robot needs to move to a parallel module at Step i for unloading the earliest wafer. Hence, in this case, y_i takes μ time units. Note that, for $\eta = (A_0 A_{i_1} \ldots A_n)$, as it can be equivalently denoted as $\eta = (A_{i_1} \ldots A_n A_0)$, A_nA_0 is an item in η. The time taken for different transitions and places is summarized in Table I.

With wafer residency time constraint, the deadlock-freeness of the obtained PN does not imply that a schedule that governs the PN is feasible. To find a feasible schedule, by [Wu *et al.*,

2008; and Wu and Zhou, 2010b], we have the following definition.

Definition 2.4 [Wu *et al.*, 2008; and Wu and Zhou, 2010b]: Given wafer residency time interval $[\alpha_i, \alpha_i+\delta_i]$ for Step i, $i \in \mathbf{N}_n$, by the PN model, a schedule is feasible if whenever u_i is enabled and fired, we have $\alpha_i \le \tau_i \le \alpha_i + \delta_i$.

III. Temporal Properties

A. Optimal Scheduling Strategy

For easy presentation and without abusing the notation, we use $A_{i-1}A_i \subseteq \eta = (A_{i_0} A_{i_1} \dots A_{i_n})$ to mean that $\exists k \in \mathbf{\Omega}_n$ such that $A_{i-1}A_i = A_{i_k} A_{i_{k+1}}$. Given a scheduling strategy $\eta = (A_{i_0} A_{i_1} \dots A_{i_n})$, define $\Lambda^P = \{i | A_{i-1}A_i \not\subseteq \eta \text{ and } i \in \mathbf{N}_n\} \cup \{0 | A_nA_0 \not\subseteq \eta\}$ and $\Lambda^R = \{i | A_{i-1}A_i \subseteq \eta \text{ and } i \in \mathbf{N}_n\} \cup \{0 | A_nA_0 \subseteq \eta\}$. For $\eta = (A_0 A_{i_1} \dots A_{i_{n-1}} A_n)$, since it is equivalent to $\eta = (A_{i_1} \dots A_{i_{n-1}} A_n A_0)$, we also have $0 \in \Lambda^R$.

The mean of Λ^P and Λ^R can be explained as follows. If $i \in \Lambda^P$, after the robot loads a wafer into Step i, it should go to another step for unloading a wafer, while if $i \in \Lambda^R$, after the robot loads a wafer into Step i, it should wait there for this wafer to be completed, instead of moving to another step for unloading a wafer if $m_i = 1$, $i \in \mathbf{N}_n$. If $0 \in \Lambda^R$, after loading a wafer into the loadlocks, the robot can unload a wafer there immediately. However, if $i \in \Lambda^R$ and meanwhile $m_i > 1$, $i \in \mathbf{N}_n$, after loading a wafer into Step i, the robot should move to a parallel module at Step i that is processing the earliest wafer that is loaded into this step for unloading it. To ease the presentation, basic cycles [Dawande *et al.*, 2002, and Chan *et al.*, 2011] are defined as follows.

Definition 3.1: For a single-robot-arm cluster tool, a strategy whose activity indexes are in a decreasing order except those in Λ^R forms a basic cycle.

For example, $\eta = (A_0A_1A_2A_3A_4)$ forms a basic cycle, since it is equivalent to $A_3A_4A_0A_1A_2$ and $\Lambda^R = \{1, 2, 4\}$ such that the condition in Definition 3.1 is met. However, $\eta = (A_0A_2A_1A_4A_3)$ does not form a basic cycle. By [Dawande *et al.*, 2002], a basic cycle dominates all other one-wafer cycle in terms of having shorter cycle time.

With only a single machine for a step and n machines for the system, Dawande *et al.* [2002] define the set D_s of machines where the processing time α_i is larger than or equal to the robot travel time from one PM to another which takes μ time units, i.e., $D_s = \{i \in \mathbf{N}_n: \alpha_i \ge \mu\}$. Then, they derive the following result.

Lemma 3.1 [Dawande *et al.* 2002]: For the following cases,

a): If $|D_s| = 0$, $(A_0A_1A_2A_3\dots A_n)$ is the optimal one-wafer scheduling strategy;

b): If $|D_s| \ge 2$, there exists an optimal basic cycle in which, $\forall i \in D_s$, we have $i \in \Lambda^P$;

c): If $|D_s| = 1$, let $\alpha_q \ge \mu$ and $\alpha_k = \max\{\alpha_j: j \in n\backslash\{q\}\}$. If $\alpha_q + \alpha_k \le 2\mu$, then $(A_0A_1A_2A_3\dots A_n)$ is the optimal one-wafer scheduling strategy. If $\alpha_q + \alpha_k > 2\mu$, there is an optimal basic one-wafer cycle in which $q \in \Lambda^P$.

In Lemma 3.1, for case a, for each step, we have $\alpha_i < \mu$. Thus, the optimal scheduling strategy is that, after loading a wafer into one step, the robot waits there until the wafer finishes its processing and then unloads it to the next step, or $(A_0A_1A_2A_3\dots A_n)$ is optimal. For case b, if $|D_s| \ge 2$, then, for each step i, $i \in D_s$, after loading a wafer into it, the robot should not wait there, instead, it should go to another step for unloading, or we have $i \in \Lambda^P$. For case c, if $|D_s| = 1$, let Step q denote the step where $\alpha_q \ge \mu$. Then, for the remaining steps, let $\alpha_k = \max\{\alpha_j: j \in n\backslash\{q\}\}$. If $\alpha_q + \alpha_k > 2\mu$, after loading a wafer into step q, the robot should go to another step for unloading, or $q \in \Lambda^P$. Otherwise, if $\alpha_q + \alpha_k \le 2\mu$, $(A_0A_1A_2A_3\dots A_n)$ is optimal.

Assume that there are parallel PMs in Step i and $i \in \Lambda^R$. After loading a wafer into Step i, the robot moves to the PM in Step i that is processing the earliest wafer loaded into this step for unloading this wafer, which takes μ time units. Further, if this wafer has not been completed yet, the robot should wait for ω_i time units. Hence, if $m_i > 1$, by treating Step i as a whole, we have that $i \in D_s$. Then, considering no wafer residency time constraint, by following Dawande *et al.* [2002], we can analyze the optimal scheduling strategy for a single-robot-arm cluster tool with parallel modules as follows:

Case 1: There is no parallel PM. For such case, we can find an optimal scheduling strategy as Dawande *et al.* [2002] do.

Case 2: There are parallel PMs and $\exists k \in \{i | m_i = 1\}$ with $\alpha_k \ge \mu$. By assumption that, for any Step i with parallel PMs, $\alpha_k \ge \mu$ holds, leading to $|D_s| \ge 2$. Then, from Lemma 3.1, for any Step i with parallel PMs, it must have $i \in \Lambda^P$ and we can obtain an optimal scheduling strategy for the whole system by a polynomial time algorithm in [Dawande *et al.*, 2002].

Case 3: There are parallel PMs and for any $k \in \{i | m_i = 1\}$, $\alpha_k < \mu$. We have two subcases: Case 3.1: There are at least two steps with parallel PMs; and Case 3.2: there is only one step with parallel PMs. For Case 3.1, it is similar to Case 2 since $|D_s| \ge 2$. Then, for any Step i with parallel PMs, $i \in \Lambda^P$ holds and an optimal scheduling strategy for the whole system can be determined based on the method in [Dawande *et al.*, 2002].

For Case 3.2, let q denote the step with m_q parallel PMs and assume that there are m_q - 1 wafers being processed in PMs 1, 2, …, and m_q - 1, respectively. Under scheduling strategy $(A_0A_1A_2A_3\dots A_n)$, after loading a wafer into the m_qth PM, the robot moves to the 1st PM, and wait for ω_q time units if necessary. Thereafter, it unloads this wafer, moves to Step q+1, and loads the wafer into it. These activities take $2\lambda + \mu + (\omega_q + \mu)$ time units. After loading a wafer into Step q+1, activity sequence ⟨processing in Step q+1 → unloading from it →

moving to Step q+2 → loading the wafer into it⟩ is followed and it takes $2\lambda + \mu + \alpha_{q+1}$ time units. Similarly, from the loading of a wafer into Step q+2 to the loading of a wafer into Step q+3, $2\lambda + \mu + \alpha_{q+2}$ time units are taken. In this way, after a cycle, when the robot loads a wafer into the 1st PM of Step q, we conclude that such a cycle takes χ time units where

$$\chi = \sum_{i=1}^{n} \alpha_i - \alpha_q + (n+1)(2\lambda + \mu) + \mu + \omega_q . \quad (3.1)$$

Let σ_1 denote the time instant of loading a wafer into the m_qth PM at Step q. From σ_1, after m_q - 1 cycles, the robot loads a wafer into the (m_q - 1)th PM at Step q. Then, the robot moves to the m_qth PM at Step q for unloading a wafer there and we let σ_2 denote this time instant. From the above analysis, we have $\sigma_2 - \sigma_1 = (m_q - 1) \times \chi + \mu$. If $\alpha_q \le \sigma_2 - \sigma_1$, before the robot comes to the m_qth PM at Step q, the wafer in it has completed its processing, thus leading to $\omega_q = 0$. However, if $\alpha_q > \sigma_2 - \sigma_1$, the wafer in it has not been completed yet, and we have $\omega_q = \alpha_q - [(m_q - 1) \times \chi + \mu]$. Such a result can be expressed as:

$$\omega_q = \begin{cases} 0, \ if\ \alpha_q \le (m_q - 1) \times \chi + \mu; \\ \alpha_q - [(m_q - 1) \times \chi + \mu], if\ \alpha_q > (m_q - 1) \times \chi + \mu. \end{cases} \quad (3.2)$$

By (3.1)-(3.2), ω_q is decided. Specifically, we first assume that $\omega_q = \alpha_q - [(m_q - 1) \times \chi + \mu]$. Then, the value of ω_q can be obtained. Next, we check whether $\alpha_q > (m_q - 1) \times \chi + \mu$ holds or not with the obtained ω_q. If not, we reset $\omega_q = 0$, otherwise the obtained value is true. Then, similar to Lemma 3.1, we can determine the optimal scheduling strategy for Case 3.2 as follows. Let $\alpha_k = \max\{\alpha_j, j \in n\backslash\{q\}\}$. If $(\mu + \omega_q) + \alpha_k \le 2\mu$, $(A_0A_1A_2A_3\ldots A_n)$ is the optimal one-wafer scheduling strategy. If $(\mu + \omega_q) + \alpha_k \ge 2\mu$, there is an optimal basic one-wafer cycle in which $q \in \Lambda^P$. Then, by following the polynomial algorithm developed in [Dawande *et al.*, 2002], an optimal scheduling strategy for the whole system can be determined.

Up to now, we have presented how to determine the optimal scheduling strategy for the system without a wafer residency time constraint. From the above analysis, for Cases 2 and 3.1, all the steps with parallel PMs belong to Λ^P. For Case 3.2, if $(\mu + \omega_q) + \alpha_k \ge 2\mu$, we still have $q \in \Lambda^P$. However, if $(\mu + \omega_q) + \alpha_k \le 2\mu$, to check the feasibility, we only need to check whether $\sigma_2 - \sigma_1 \le \alpha_q + \delta_q$ holds or not. If yes, $(A_0A_1A_2A_3\ldots A_n)$ is an optimal and feasible schedule; otherwise the wafer in Step q violates the wafer residency constraint.

In the following discussion, we assume that an optimal scheduling strategy is determined, i.e., the robot task sequence is determined. Nevertheless, this strategy cannot guarantee that wafer residency time constraint is satisfied. Then, based on the given strategy, the key is how to find a feasible schedule. To do so, we just need to analyze Cases 1, 2, 3.1 and the situation where $(\mu + \omega_q) + \alpha_k \ge 2\mu$ for Case 3.2. From the above analysis, we have that, for all of the cases, any step with parallel modules belongs to Λ^P.

B. Temporal Aspect Analysis

According to Chan *et al.* [2011], given strategy $\eta = (A_{i_0} A_{i_1} \ldots A_{i_n})$, a wafer processing cycle at Step i is formed by the activity sequence from A_i to A_{i-1}, i.e., $(A_i A_{j_1} A_{j_2} \ldots A_{j_m} A_{i-1})$, where $(j_1, j_2, \ldots j_m)$ are the indexes of activities between i and i - 1 under η. For example, given $\eta = (A_0A_5A_3A_4A_1A_2)$, $A_1A_2A_0$ forms such a cycle at Step 0, while $A_5A_3A_4$ forms a cycle at Step 5. In a robot cycle, all activities in η should be performed once.

To analyze such cycles, as done in [Chan *et al.*, 2011], we use $\bigcap_i$ and $\bigcap_R$ to denote the index sets of activities in the wafer processing cycle at Step i and robot cycle, respectively, i.e., $\bigcap_i = \{i, j_1, j_2, \ldots j_m, i - 1\}$ and $\bigcap_R = \{i_0, i_1, i_2, \ldots i_n\}$. Furthermore, for $i \in \mathbf{\Omega_n}$, let $\omega_i = \omega_i^* + \alpha_i$ if $i \in \Lambda^R$, and $\omega_i = \omega_i^*$ if $i \in \Lambda^P$.

Then, we analyze the time taken for a wafer at Step i. To satisfy the wafer residency time constraint at Step i, $\tau_i \in [\alpha_i, \alpha_i + \delta_i]$, $i \in \mathbf{N_n}$, must hold. When $\tau_i = \alpha_i$, the allowed shortest wafer processing cycle time at Step i is reached and we use θ_i to denote it; while $\tau_i = \alpha_i + \delta_i$, the allowed longest time to complete a wafer at Step i is reached and we use Θ_i to denote it. We first discuss how to calculate θ_i.

If $|\bigcap_i| = 2$, we have $\bigcap_i = \{i, i - 1\}$, $i \in \mathbf{N_n}$, or $\bigcap_0 = \{0, n\}$. In this case, to complete a wafer at Step i, $i \in \mathbf{N_n}$, task sequence ⟨processing a wafer at Step i (α_i) → firing u_i (λ) → x_{i+1} (μ) → t_{i+1} (λ) → y_{i-1} (μ) → waiting in q_{i-1} (ω_{i-1}^*) → u_{i-1} (λ) → x_i (μ) → firing t_i (λ)⟩ is performed, which takes $\alpha_i + 4\lambda + 3\mu + \omega_{i-1}^*$ time units. Considering that there are m_i parallel PMs at Step i and all the steps with parallel PMs belong to Λ^P, we have $\theta_i = (\alpha_i + 4\lambda + 3\mu + \omega_{i-1}^*)/m_i$, $i \in \mathbf{N_n}$. Similarly, we have $\theta_0 = (\alpha_0 + 4\lambda + 3\mu + \omega_n^*)/m_0$, where $m_0 = 1$ is set, since the loadlocks has no wafer processing function and there is no wafer residency time constraint.

If $|\bigcap_i| = 3$, we have $\bigcap_i = \{i, k, i - 1\}$, $i \in \mathbf{N_n}$. For the case of $\bigcap_i = \{i, k, i - 1\}$, if $k \in \Lambda^R$ and $(i - 1) \in \Lambda^P$, let ς_1 denote the time instant when A_i has just been performed, i.e., t_{i+1} (or t_k) finishes its firing at ς_1. As $k \in \Lambda^R$, y_k takes no time and the robot waits for ($\alpha_k + \omega_k^*$) time units at p_k. Then, the robot unloads a wafer there by firing u_k, moves to Step k + 1 and loads the wafer into the step by firing x_{k+1} and t_{k+1}. Let ς_2 denote the time instant when t_{k+1} finishes its firing. Then, we have $\varsigma_2 - \varsigma_1 = 2\lambda + \mu + (\alpha_k + \omega_k^*)$. By starting from ς_2, activity sequence ⟨firing y_{i-1} (μ) (since $(i - 1) \in \Lambda^P$) →

waiting in q_{i-1} (ω^*_{i-1}) $\rightarrow$ u_{i-1} (λ) $\rightarrow$ x_i (μ) $\rightarrow$ t_i (λ) $\rightarrow$ processing a wafer at Step i (α_i) $\rightarrow u_i(\lambda) \rightarrow x_{i+1}(\mu) \rightarrow t_{i+1}(\lambda)\rangle$ is performed such that a wafer processing cycle at Step i is completed. Let ς_3 denote the time instant when t_{i+1} finishes its firing again. Then, $\varsigma_3 - \varsigma_2 = \alpha_i + 4\lambda + 3\mu + \omega^*_{i-1}$. Thus, we have $\theta_i = (\varsigma_3 - \varsigma_1)/m_i = (\alpha_i + 4\lambda + 3\mu + \omega^*_{i-1} + 2\lambda + \mu + \alpha_k + \omega^*_k)/m_i$.

If $k \in \Lambda^P$ and $(i - 1) \in \Lambda^R$. Let ς_1 and ς_2 denote the time instant when t_{i+1} and t_{k+1} finish their firing. As $k \in \Lambda^P$, y_k takes μ time units and we have $\varsigma_2 - \varsigma_1 = 2\lambda + 2\mu + \omega^*_k$. Let ς_3 denote the time instant when t_{i+1} finishes its firing again. As y_{i-1} takes no time for this case, from the above analysis, we have $\varsigma_3 - \varsigma_2 = \alpha_i + 4\lambda + 2\mu + \alpha_{i-1} + \omega^*_{i-1}$. Thus, we have $\theta_i = (\varsigma_3 - \varsigma_1)/m_i = (\alpha_i + 4\lambda + 2\mu + \alpha_{i-1} + \omega^*_{i-1} + 2\lambda + 2\mu + \omega^*_k)/m_i$.

By the above analysis, if one basic activity A_k is added, θ_i increases $(2\lambda + 2\mu + \omega_k)/m_i$ time units, where $\omega_i = \omega^*_i + \alpha_i$ if $i \in \Lambda^R$, and $\omega_i = \omega^*_i$ if $i \in \Lambda^P$. Meanwhile, in $\bigcap_i \backslash \{i\}$, if the number of indexes that belong to Λ^R increases by one, θ_i decreases μ/m_i time units. Hence, we can calculate θ_i for the cases of $|\bigcap_i| \in \{4, 5, 6, \ldots, n+1\}$. To ease the presentation, let $Q_i = \{\bigcap_i \backslash \{i\} \cap \Lambda^R\}$ and $R_i = \{\bigcap_i \backslash \{i\} \cap \Lambda^P\}$. Then, we have the allowed shortest cycle time at Step i is:

$$\theta_i = (\alpha_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + (\sum_{k \in Q_i} (\alpha_k + \omega^*_k) + \sum_{k \in R_i} \omega^*_k) - |Q_i| \times \mu)/m_i. \quad (3.3)$$

The longest wafer processing cycle time is:

$$\Theta_i = (\alpha_i + \delta_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + (\sum_{k \in Q_i} (\alpha_k + \omega^*_k) + \sum_{k \in R_i} \omega^*_k) - |Q_i| \times \mu)/m_i. \quad (3.4)$$

Expressions (3.3)-(3.4) present the feasible cycle time range or workload range for Step i, $i \in \mathbf{N}_n$. By removing $\sum_{k \in Q_i} \omega^*_k$ and $\sum_{k \in R_i} \omega^*_k$ from (3.3)-(3.4), we have the shortest and longest "natural workload" [Wu *et al*., 2008] for Step i as follows:

$$\xi_i = (\alpha_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k \in Q_i} \alpha_k - |Q_i| \times \mu)/m_i. \quad (3.5)$$

$$\Xi_i = (\alpha_i + \delta_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k \in Q_i} \alpha_k - |Q_i| \times \mu)/m_i. \quad (3.6)$$

Natural workload plays an important role in schedulability analysis, since it provides the workload balance information among the steps. Let $\theta^*_i = (\tau_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k \in \{I_i \backslash \{i\}\}} \omega_k - |Q_i| \times \mu)/m_i$, $i \in \mathbf{\Omega}_n$, denote the cycle time for Step i. By comparing θ^*_i with $\xi^*_i = (\tau_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k \in Q_i} \alpha_k - |Q_i| \times \mu)/m_i$, we conclude that if $\theta^*_i = \xi^*_i$, the τ_i in ξ^*_i is greater than or equal to that in θ^*_i, since $\sum_{k \in \{I_i \backslash \{i\}\}} \omega_k = \sum_{k \in Q_i} \alpha_k + \sum_{k \in Q_i} \omega^*_k + \sum_{k \in R_i} \omega^*_k \geq \sum_{k \in Q_i} \alpha_k$. This implies that, by adjusting the robot waiting time, we can schedule the system to balance the workloads among the steps to some extent to make the wafer residency time constraint satisfied. To do so, we need to know how to calculate τ_i that is dependent on the robot cycle time ψ. Now, we discuss how to calculate the robot cycle time ψ.

Let ψ^* be the robot cycle time under the assumption that $\omega^*_k = 0$, $k \in \mathbf{\Omega}_n$. If it is a backward strategy, by [Wu *et al*., 2008] we have:

$$\psi = 2(n + 1)(\lambda + \mu) + \sum_{i=0}^{n} \omega_i = \psi^* + \sum_{i=0}^{n} \omega_i \ . \quad (3.7)$$

If it is not the backward one, by [Dawande *et al*., 2002], there must be at least one $i \in \Lambda^R$. Then, by [Chan *et al*., 2011], ψ^* is equal to ξ_i, $i \in \Lambda^R$, or we have:

$$\psi^* = (\alpha_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) - |Q_i| \times \mu + \sum_{k \in Q_i} \alpha_k)/m_i, \ i \in \Lambda^R. \quad (3.8)$$

Let $\Pi = \max\{\xi_0, \xi_1, \xi_2, \ldots, \xi_n, \psi^*\}$. Then, Π must be the lower bound of cycle time for the system. With cycle time Π, if we can find a feasible schedule, it must be optimal. Thus, in this paper, we let Π be the cycle time of the system.

With θ^*_i, $i \in \mathbf{\Omega}_n$, being the cycle time of Step i, by [Wu *et al*., 2008], in the steady state, each step has the same cycle time that should be equal to the robot cycle time ψ and the system cycle time, or we have:

$$\Pi = \theta^*_0 = \theta^*_1 = \ldots = \theta^*_n = \psi. \quad (3.9)$$

By (3.9), we have:

$$\psi = \Pi = \theta^*_i = (\tau_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k \in \{I_i \backslash \{i\}\}} \omega_k - |Q_i| \times \mu)/m_i, \ i \in \mathbf{\Omega}_n. \quad (3.10)$$

By (3.10), in the steady-state, the wafer sojourn time τ_i in p_i, $i \in \mathbf{N}_n$, is:

$$\tau_i = m_i \times \Pi - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k \in \{I_i \backslash \{i\}\}} \omega_k - |Q_i| \times \mu]. \quad (3.11)$$

With (3.11), once ω_i, $i \in \mathbf{\Omega}_n$, is determined, τ_i, $i \in \mathbf{N}_n$, can be calculated. Then, we discuss the existence of a feasible

schedule and how to find it if existing next.

IV. Schedulability and Scheduling

It follows from (3.9) that to schedule the system is to allocate the robot waiting time such that the wafer residency time constraint is satisfied, or the scheduling a schedule is parameterized by the robot waiting time. Then, we present the following Lemma.

Lemma 4.1: For a single-robot-arm tool with wafer residency time constraint, if $\psi^* = \Pi \le \Xi_i$, $i \in \mathbf{N_n}$, for $k \in \mathbf{\Omega_n}$, set $\omega_k = \alpha_k$ if $k \in \Lambda^R$, otherwise $\omega_k = 0$, then, a feasible one-wafer cyclic schedule can be obtained, or $\alpha_i \le \tau_i \le \alpha_i + \delta_i$, $i \in \mathbf{N_n}$, holds.

Proof: If $\psi^* = \Pi \le \Xi_i$, $i \in \mathbf{N_n}$, we have $\omega_k^* = 0$, $k \in \mathbf{\Omega_n}$ such that $\omega_k = \omega_k^* + \alpha_k = \alpha_k$, $k \in \Lambda^R$; and $\omega_k = \omega_k^* = 0$, $k \in \Lambda^P$. Hence, we have $\sum_{k\in\{I_i\setminus\{i\}\}} \omega_k = \sum_{k\in Q_i} \alpha_k$.

For $i \in \Lambda^R$, by $\omega_i = \alpha_i$, it means that a wafer is unloaded immediately after it is completed, or the wafer residency time constraint is met. For $i \in \Lambda^P$, it follows from $\Pi = \max\{\xi_0, \xi_1, \xi_2, \ldots, \xi_n, \psi^*\} = \psi^* \le \Xi_i$ and (3.11) that we have $\tau_i \ge m_i \times \xi_i - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}} \omega_k - |Q_i| \times \mu] = (\alpha_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in Q_i} \alpha_k - |Q_i| \times \mu) - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}} \omega_k - |Q_i| \times \mu] = \alpha_i$; and $\tau_i \le m_i \times \Xi_i - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}} \omega_k - |Q_i| \times \mu] = (\alpha_i + \delta_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in Q_i} \alpha_k - |Q_i| \times \mu) - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}} \omega_k - |Q_i| \times \mu] = \alpha_i + \delta_i$. In other words, we have $\alpha_i \le \tau_i \le \alpha_i + \delta_i$, $i \in \mathbf{N_n}$. ■

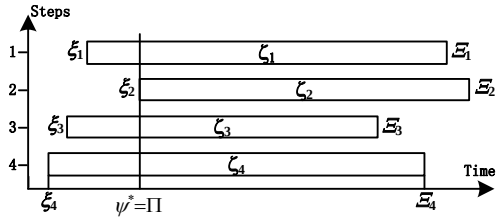


Fig 3. Illustration of Lemma 4.1.

The situation given in Lemma 4.1 can be illustrated in Fig. 3, where ζ_i presents the permissive value in $[\xi_i, \Xi_i]$, $i \in \mathbf{N_n}$. Notice that ζ_i, ψ^*, and Π stand for time durations but not for time points. In Fig. 3, $\max\{\xi_0, \xi_1, \xi_2, \ldots, \xi_n, \psi^*\} = \psi^* = \Pi \le \Xi_i$. Then, by setting $\omega_k = \omega_k^* = \alpha_k$, $k \in \Lambda^R$; and $\omega_k = \omega_k^* = 0$, $k \in \Lambda^P$, the wafer residency time constraint is satisfied for every Step i, $i \in \mathbf{N_n}$.

By $\Pi = \max\{\xi_0, \xi_1, \xi_2, \ldots, \xi_n, \psi^*\}$, we have another case when $\psi^* < \Pi \le \Xi_i$. In this case, to find a one-wafer cyclic schedule, the key is how to allocate $(\Pi - \psi^*)$ to ω_k^*'s, $k \in \mathbf{\Omega_n}$. With Π being the cycle time, for Step j, $j \in \mathbf{\Omega_n}$, there are at most $m_j \times (\Pi - \xi_j)$ time units that can be allocated to $\sum_{k\in(I_j\setminus\{j\})} \omega_k^*$, or we have Case 1): $\sum_{k\in(I_j\setminus\{j\})} \omega_k^* \le m_j \times (\Pi - \xi_j)$, $j \in \mathbf{\Omega_n}$. By (3.7)-(3.9), we have Case 2): $\sum_{k\in\Omega_n} \omega_k^* = \Pi - \psi^*$; and Case 3): $\omega_k^* \ge 0$, $k \in \mathbf{\Omega_n}$. Then, the following linear program is proposed to determine ω_k^*, $k \in \mathbf{\Omega_n}$.

$$Min = \omega_0^*$$
$$\begin{cases} \sum_{k\in(I_j\setminus\{j\})} \omega_k^* \le m_j \times (\Pi - \xi_j),\ j \in \Omega_n \\ \sum_{k\in\Omega_n} \omega_k^* = \Pi - \psi^* \\ \omega_k^* \ge 0,\ k \in \Omega_n \end{cases} \tag{4.1}$$

This linear program can be efficiently solved due to its limited size. With ω_k^* being determined by (4.1), we present the following Lemma.

Lemma 4.2: For a single-robot-arm tool with wafer residency time constraint, if $\psi^* < \Pi \le \Xi_i$, $i \in \mathbf{N_n}$, determine ω_k^* by solving Problem (4.1), and thereafter set $\omega_k = \omega_k^* + \alpha_k$ if $k \in \Lambda^R$, otherwise $\omega_k = \omega_k^*$, then, a feasible one-wafer cyclic schedule can be obtained, or $\alpha_i \le \tau_i \le \alpha_i + \delta_i$, $i \in \mathbf{N_n}$, holds.

Proof: If $\psi^* < \Pi \le \Xi_i$, $i \in \mathbf{N_n}$, by Lemma 4.2, we have $\omega_k = \omega_k^* + \alpha_k$, $k \in \Lambda^R$; and $\omega_k = \omega_k^*$, $k \in \Lambda^P$. Obviously, $\sum_{k\in Q_i} \alpha_k + \sum_{k\in\{I_i\setminus\{i\}\}} \omega_k^* = \sum_{k\in\{I_i\setminus\{i\}\}} \omega_k$ holds.

By Problem (4.1), $\xi_i + (\sum_{k\in\{I_i\setminus\{i\}\}} \omega_k^*)/m_i \le \Pi$, $i \in \mathbf{N_n}$. Hence, it follows from (3.11) that, $\tau_i \ge m_i \times [\xi_i + (\sum_{k\in\{I_i\setminus\{i\}\}} \omega_k^*)/m_i] - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}} \omega_k - |Q_i| \times \mu] = m_i \times \xi_i + \sum_{k\in(I_i\setminus\{i\})} \omega_k^* - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}} \omega_k - |Q_i| \times \mu] = [\alpha_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in Q_i} \alpha_k - |Q_i| \times \mu]$

$+ \sum_{k\in(I_i\setminus\{i\})}\omega_k^* - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}}\omega_k - |Q_i| \times \mu] = \alpha_i$.

As $\Pi \le \Xi_i$, $i \in \mathbf{N}_n$, it follows from (3.11) that, $\tau_i \le m_i \times \Xi_i - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}}\omega_k - |Q_i| \times \mu] = [\alpha_i + \delta_i + 4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in Q_i}\alpha_k - |Q_i| \times \mu] - [4\lambda + 3\mu + 2 \times (|\bigcap_i| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_i\setminus\{i\}\}}\omega_k - |Q_i| \times \mu] = \alpha_i + \delta_i + \sum_{k\in Q_i}\alpha_k - \sum_{k\in\{I_i\setminus\{i\}\}}\omega_k = \alpha_i + \delta_i - \sum_{k\in(I_i\setminus\{i\})}\omega_k^*$.

Thus, to prove $\tau_i \le \alpha_i + \delta_i$, we need to prove that $0 \le \delta_i - \sum_{k\in(I_i\setminus\{i\})}\omega_k^* \le \delta_i$ holds. Since $\sum_{k\in(I_i\setminus\{i\})}\omega_k^* \ge 0$, we have $\delta_i - \sum_{k\in(I_i\setminus\{i\})}\omega_k^* \le \delta_i$. Then, we need to show $\delta_i - \sum_{k\in(I_i\setminus\{i\})}\omega_k^* \ge 0$. From the above analysis, we have ①: $\xi_i \le \Pi - (\sum_{k\in(I_i\setminus\{i\})}\omega_k^*)/m_i \le \Xi_i - (\sum_{k\in(I_i\setminus\{i\})}\omega_k^*)/m_i$. Also, from (3.5)-(3.6), we have ②: $\xi_i + \delta_i/m_i = \Xi_i$. By Combining ① and ②, we have $\xi_i \le \xi_i + \delta_i/m_i - (\sum_{k\in(I_i\setminus\{i\})}\omega_k^*)/m_i$, i.e., $0 \le (\delta_i - \sum_{k\in(I_i\setminus\{i\})}\omega_k^*)/m_i$, or $\delta_i \ge \sum_{k\in(I_i\setminus\{i\})}\omega_k^*$. Hence, we always have $\alpha_i \le \tau_i \le \alpha_i + \delta_i$, $i \in \mathbf{N}_n$ holds. ■

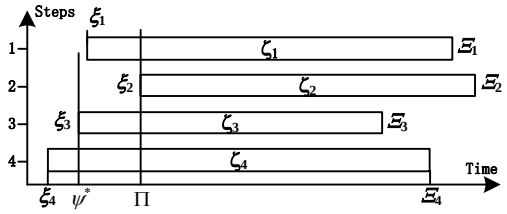


Fig 4. Illustration of Lemma 4.2.

Similarly, the situation given in Lemma 4.2 is illustrated in Fig. 4.

In Lemmas 4.1-4.2, it requires that $\Pi \le \Xi_i$, $i \in \mathbf{N}_n$. However, $\Xi_i < \Pi$ may hold for at least one $i \in \mathbf{N}_n$. In this case, let $\mathbf{E} = \{j | j \in \mathbf{N}_n, \Xi_j < \Pi\}$ and $\mathbf{F} = \mathbf{\Omega}_n \setminus \mathbf{E}$. For $j \in \mathbf{E}$, to make the wafer residency time constraint satisfied, at least $m_j \times (\Pi - \Xi_j)$ time units should be allocated to $\sum_{k\in(I_j\setminus\{j\})}\omega_k^*$, or we have Case 1): $m_j \times (\Pi - \Xi_j) \le \sum_{k\in(I_j\setminus\{j\})}\omega_k^*$, $j \in \mathbf{E}$; for $j \in \mathbf{\Omega}_n$, similar to (4.1), we have Case 2): $\sum_{k\in(I_j\setminus\{j\})}\omega_k^* \le m_j \times (\Pi - \xi_j)$; also similar to (4.1), we have Case 3): $\sum_{k\in\Omega_n}\omega_k^* = \Pi - \psi^*$, and Case 4): $\omega_k^* \ge 0$. Then, the following linear program is proposed to find ω_k^*, $k \in \mathbf{\Omega}_n$.

$$Min = \omega_0^*$$
$$\begin{cases} m_j \times (\Pi - \Xi_j) \le \sum_{k\in(I_j\setminus\{j\})}\omega_k^*, \ j \in E \\ \sum_{k\in(I_j\setminus\{j\})}\omega_k^* \le m_j \times (\Pi - \xi_j), \ j \in \Omega_n \\ \sum_{k\in\Omega_n}\omega_k^* = \Pi - \psi^* \\ \omega_k^* \ge 0, \ k \in \Omega_n \end{cases} \quad (4.2)$$

Then, we have the following Lemma.

Lemma 4.3: For a single-robot-arm tool with wafer residency time constraint, if $j \in \mathbf{E} \ne \varnothing$ and $\partial = \sum_{j\in E} m_j \times (\Pi - \Xi_j) \le \Pi - \psi^*$, determine ω_k^* by solving Problem (4.2), and thereafter set $\omega_k = \omega_k^* + \alpha_k$ if $k \in \Lambda^R$, otherwise $\omega_k = \omega_k^*$, then, a feasible one-wafer cyclic schedule can be obtained, or $\alpha_i \le \tau_i \le \alpha_i + \delta_i$, $i \in \mathbf{N}_n$, holds.

Proof: For $j \in \mathbf{F}$, we have $\psi^* < \max\{\xi_0, \xi_1, \xi_2, \ldots, \xi_n, \psi^*\} = \Pi \le \Xi_j$. It follows from Lemma 4.2 that $\alpha_j \le \tau_j \le \alpha_j + \delta_j$, $j \in$ F, holds.

For $j \in \mathbf{E}$, by Problem (4.2), we have $(\sum_{k\in(I_j\setminus\{j\})}\omega_k^*)/m_j + \xi_j \le \Pi \le (\sum_{k\in(I_j\setminus\{j\})}\omega_k^*)/m_j + \Xi_j$. By Lemma 4.3, we set $\omega_k = \omega_k^* + \alpha_k$, $k \in \Lambda^R$; and $\omega_k = \omega_k^*$, $k \in \Lambda^P$; resulting in $\sum_{k\in Q_j}\alpha_k + \sum_{k\in(I_j\setminus\{j\})}\omega_k^* = \sum_{k\in\{I_j\setminus\{j\}\}}\omega_k$. With $(\sum_{k\in(I_j\setminus\{j\})}\omega_k^*)/m_j + \xi_j \le \Pi$, similar to Lemma 4.2, we can show that $\tau_j \ge \alpha_j$. With $\Pi \le (\sum_{k\in(I_j\setminus\{j\})}\omega_k^*)/m_j + \Xi_j$, from (3.11), we have $\tau_j \le m_j \times [(\sum_{k\in(I_j\setminus\{j\})}\omega_k^*)/m_j + \Xi_j] - [4\lambda + 3\mu + 2 \times (|\bigcap_j| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_j\setminus\{j\}\}}\omega_k - |Q_j| \times \mu] = \sum_{k\in(I_j\setminus\{j\})}\omega_k^* + [\alpha_j + \delta_j + 4\lambda + 3\mu + 2 \times (|\bigcap_j| - 2) \times (\lambda + \mu) + \sum_{k\in Q_j}\alpha_k - |Q_j| \times \mu] - [4\lambda + 3\mu + 2 \times (|\bigcap_j| - 2) \times (\lambda + \mu) + \sum_{k\in\{I_j\setminus\{j\}\}}\omega_k - |Q_j| \times \mu] = \alpha_j + \delta_j$. In other words, $\alpha_j \le \tau_j \le \alpha_j + \delta_j$, $j \in \mathbf{N}_n$, holds. ■

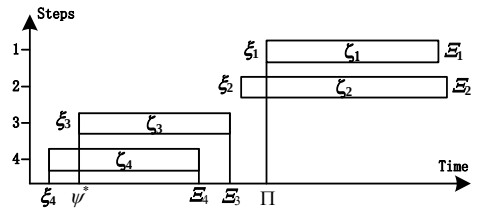


Fig 5. Illustration of Lemma 4.3.

The situation given in Lemma 4.3 is illustrated in Fig. 5. In Lemma 4.3, we assume that $\partial = \sum_{j\in E} m_j \times (\Pi - \Xi_j) \le \Pi - \psi^*$. If this assumption is not true, it implies that there is not enough robot waiting time to be assigned to $\sum_{k\in(\mathrm{I}_j\setminus\{j\})} \omega_k^*$ such that $\Xi_j \ge \Pi$, for $\forall j \in \mathbf{E}$. In this case, there is no feasible one-wafer cyclic schedule for the system. To conclude the scheduling conditions, we have the following Theorem.

Theorem 4.1: For a single-robot-arm cluster tool, an optimal and feasible one-wafer cyclic schedule can be obtained if and only if the conditions given in one of Lemmas 4.1 - 4.3 hold. Or the conditions in Lemmas 4.1-4.3 are necessary and sufficient.

Proof: From (3.9)-(3.10) and Lemmas 4.1-4.3 we know that the schedule is parameterized by the robot waiting time, furthermore, the robot waiting time is fixed all the time. Therefore, it can be easily got that during each cycle, only one wafer is unloaded from or loaded into the loadlocks, hence, it is a one-wafer cyclic schedule. Notice that Π is the lower bound of cycle time for the system without wafer residency time constraint. Thus, the obtained schedule must be optimal in the sense of cycle time.

As $\Pi = \max\{\xi_0, \xi_1, \xi_2, \ldots, \xi_n, \psi^*\}$, hence, $\psi^* \le \Pi$ always holds. Note that there is no wafer residency time constraint in the loadloakcs or Step 0. Thus, for $i \in \mathbf{N}_n$, if $\Pi \le \Xi_i$, we have two different cases. One is $\psi^* = \Pi \le \Xi_i$, $i \in \mathbf{N}_n$, and the other is $\psi^* < \Pi \le \Xi_i$, $i \in \mathbf{N}_n$, which are stated in Lemmas 4.1 and 4.2.

If $\Xi_i < \Pi$ satisfied for at least one step i, $i \in \mathbf{N}_n$, we also have two different cases. One is $\partial = \sum_{j\in E} m_j \times (\Pi - \Xi_j) \le \Pi - \psi^*$, and the other is $\partial = \sum_{j\in E} m_j \times (\Pi - \Xi_j) > \Pi - \psi^*$. For the former, we have proved in Lemma 4.3 that it meets the wafer residency time constraint. And for the latter, from the above analysis we know that there is no feasible one-wafer cyclic schedule. Thus, the conditions in Lemmas 4.1-4.3 are necessary and sufficient regarding the existence of an optimal and feasible one-wafer cyclic schedule for single-robot-arm cluster tools. ■

Next, we need to analyze the computational complexity for finding a feasible schedule.

First, we need to determine the optimal scheduling strategy for the system, which can be done by applying the polynomial time algorithm derived in [Dawande *et al.*, 2002]. Since the obtained strategy determines the robot task sequence only, it cannot guarantee the feasibility. Based on the obtained strategy, a feasible schedule is found by determining the robot waiting time if such a schedule exists. To do so, in the worst case, a linear program (4.1) or (4.2) is solved to determine the value of ω_k^*, $k \in \mathbf{\Omega}_n$. It is well-known that a linear program can be efficiently solved by a commercial solver. Hence, the proposed method is computationally efficient.

V. Illustrative Examples

In this section, two industrial examples are used to show the application of the proposed method.

Example 1: In a cluster tool, there are three processing steps. Steps 1 and 3 consist of just 1 process module (PM), and Step 2 has two PMs, or we have $(m_1, m_2, m_3) = (1, 2, 1)$. Next, we consider three cases.

Case 1: It takes 50, 116, and 6 time units for a PM at Steps 1, 2, and 3 to process a wafer, respectively; 3 time units for the robot to unload/load a wafer from/into a step; 10 time units for the robot to move between PMs with or without holding a wafer. After being processed, a wafer can stay at each step for at most 20 time units. In other words, we have $\alpha_1 = 50$, $\alpha_2 = 116$, $\alpha_3 = 6$, $\lambda = 3$, $\mu = 10$, and $\delta_1 = \delta_2 = \delta_3 = 20$.

For this case, by the method in [Dawande *et al.*, 2002], the optimal scheduling strategy is found to be $(A_0A_2A_3A_1)$. By (3.5), the shortest "natural workload" at each step is that:

$\xi_0 = \alpha_3 + 6\lambda + 4\mu = 6 + 18 + 40 = 64$,

$\xi_1 = \alpha_1 + 4\lambda + 3\mu = 50 + 12 + 30 = 92$,

$\xi_2 = (\alpha_2 + \alpha_3 + 6\lambda + 4\mu)/2 = (116 + 6 + 18 + 40)/2 = 90$, and

$\xi_3 = \alpha_3 + 8\lambda + 7\mu = 6 + 24 + 70 = 100$.

With ψ^* denoting the robot cycle time under the assumption that $\omega_k^* = 0$, $k \in \mathbf{\Omega}_n$, by (3.8) we have $\psi^* = \xi_3 = 100$.

Hence, the lower bound of cycle time for the system (denoted by Π) is that: $\Pi = \max\{\xi_0, \xi_1, \xi_2, \xi_3, \psi^*\} = 100$.

By (3.6), the longest "natural workload" at each step is that:

$\Xi_1 = \xi_1 + 20 = 112$,

$\Xi_2 = \xi_2 + 20/2 = 100$, and

$\Xi_3 = \xi_3 + 20 = 120$.

Thus, we have $\psi^* = \Pi = 100 \le \Xi_i$, $i \in \mathbf{N}_3$, and the conditions given in Lemma 4.1 are satisfied. Then, by Lemma 4.1, we set $\omega_0 = \omega_0^* = 0$, $\omega_1 = \omega_1^* = 0$, $\omega_2 = \omega_2^* = 0$, and $\omega_3 = \omega_3^* + \alpha_3 = 6$. In such a way, an optimal and feasible one-wafer cyclic schedule is obtained.

Case 2: In Case 1, the wafer processing time for Step 2 is changed to $\alpha_2 = 140$, and the other parameters are unchanged.

In this case, by [Dawande *et al.*, 2002], the optimal scheduling strategy is still $(A_0A_2A_3A_1)$. With this strategy, we have:

$\xi_0 = \alpha_3 + 6\lambda + 4\mu = 64$,

$\xi_1 = \alpha_1 + 4\lambda + 3\mu = 92$,

$\xi_2 = (\alpha_2 + \alpha_3 + 6\lambda + 4\mu)/2 = (140 + 6 + 18 + 40)/2 = 102$,

$\xi_3 = \alpha_3 + 8\lambda + 7\mu = 100 = \psi^*$, and

$\Pi = \max\{\xi_0, \xi_1, \xi_2, \xi_3, \psi^*\} = 102$.

By (3.6), we have:

$\Xi_1 = \xi_1 + 20 = 112$,

$\Xi_2 = \xi_2 + 20/2 = 112$, and

$\Xi_3 = \xi_3 + 20 = 120$.

Hence, $\psi^* = 100 < \Pi = 102 \le \Xi_i$, $i \in \mathbf{N_3}$, holds, i.e., the conditions in Lemma 4.2 are satisfied. According to Problem (4.1), we have linear program (4.3).

$$Min = \omega_0^*$$
$$\begin{cases} \omega_2^* + \omega_3^* \le 38 \\ \omega_0^* \le 10 \\ \omega_1^* + \omega_3^* \le 0 \\ \omega_0^* + \omega_1^* + \omega_2^* \le 2 \\ \omega_0^* + \omega_1^* + \omega_2^* + \omega_3^* = 2 \\ \omega_0^*, \omega_1^*, \omega_2^*, \omega_3^* \ge 0 \end{cases} \quad (4.3)$$

By solving (4.3), we have $\omega_0^* = \omega_1^* = \omega_3^* = 0$, and $\omega_2^* = 2$. Then, by Lemma 4.2, we set $\omega_0 = \omega_0^* = 0$, $\omega_1 = \omega_1^* = 0$, $\omega_2 = \omega_2^* = 2$, and $\omega_3 = \omega_3^* + \alpha_3 = 6$ such that an optimal and feasible one-wafer cyclic schedule is found.

Case 3: In Case 1, the wafer processing time for Steps 1 and 2 are changed as $\alpha_1 = 40$, and $\alpha_2 = 140$, δ_i's, $i \in \mathbf{N_3}$, are changed as $\delta_1 = \delta_2 = \delta_3 = 16$, and the others are unchanged.

In this case, by [Dawande *et al.*, 2002] we can find that the optimal scheduling strategy is $(A_0A_2A_3A_1)$. Then, we have:

$\xi_0 = \alpha_3 + 6\lambda + 4\mu = 64$,
$\xi_1 = \alpha_1 + 4\lambda + 3\mu = 40 + 12 + 30 = 82$,
$\xi_2 = (\alpha_2 + \alpha_3 + 6\lambda + 4\mu)/2 = (140 + 6 + 18 + 40)/2 = 102$,
$\xi_3 = \alpha_3 + 8\lambda + 7\mu = 100 = \psi^*$, and
$\Pi = \max\{\xi_0, \xi_1, \xi_2, \xi_3, \psi^*\} = 102$.

By (3.6), we have:
$\Xi_1 = \xi_1 + 16 = 98 < \Pi = 102$,
$\Xi_2 = \xi_2 + 16/2 = 110 > \Pi$, and
$\Xi_3 = \xi_3 + 16 = 116 > \Pi$.

Hence, by the definition of $\mathbf{E} = \{j | j \in \mathbf{N}_n, \Xi_j < \Pi\}$ and $\mathbf{F} = \mathbf{\Omega}_n \backslash \mathbf{E}$, we have $1 \in \mathbf{E}$ and $2, 3 \in \mathbf{F}$. Since $\partial = \sum_{j \in E} m_j \times (\Pi - \Xi_j) = \Pi - \Xi_1 = 102 - 98 = 4 > \Pi - \psi^* = 102 - 100 = 2$, by Theorem 4.1 we know that there is no feasible cyclic schedule.

Example 2: In this example, Steps 1 and 2 are configured with one PM, and Step 3 with two PMs, or $(m_1, m_2, m_3) = (1, 1, 2)$. And it takes 20, 30, and 200 time units for a PM at Steps 1, 2, and 3 to process a wafer, respectively; 8 time units for the robot to unload/load a wafer from/into a step; 2 time units for the robot to move between PMs with or without carrying a wafer; and after being processed, a wafer at each step can stay there for at most 10 time units. Or we have $\alpha_1 = 20$, $\alpha_2 = 30$, $\alpha_3 = 200$, $\lambda = 8$, $\mu = 2$, and $\delta_1 = \delta_2 = \delta_3 = 10$.

Based on [Dawande *et al.*, 2002], we have two optimal scheduling strategies for this case. One is $(A_0A_1A_3A_2)$ and the other is $(A_0A_3A_2A_1)$, or the backward strategy.

With $(A_0A_3A_2A_1)$, we have:
$\xi_0 = \alpha_0 + 4\lambda + 3\mu = 0 + 4 \times 8 + 3 \times 2 = 38$,
$\xi_1 = \alpha_1 + 4\lambda + 3\mu = 58$,
$\xi_2 = \alpha_2 + 4\lambda + 3\mu = 68$, and
$\xi_3 = (\alpha_3 + 4\lambda + 3\mu)/2 = 119$.

Based on [Wu *et al.*, 2008], we have $\psi^* = 2(n + 1) \times (\lambda + \mu) = 8 \times 10 = 80$. Hence, $\Pi = \max\{\xi_0, \xi_1, \xi_2, \xi_3, \psi^*\} = 119$. By (3.6), we have:
$\Xi_1 = \xi_1 + \delta_1 = \xi_1 + 10 = 68 < \Pi = 119$,
$\Xi_2 = \xi_2 + 10 = 78 < \Pi = 119$, and
$\Xi_3 = \xi_3 + \delta_3/2 = 119 + 5 > \Pi = 119$.

Hence, we have $1, 2 \in \mathbf{E}$ and $3 \in \mathbf{F}$. Since $\partial = \sum_{j \in E} m_j \times (\Pi - \Xi_j) = (\Pi - \Xi_1) + (\Pi - \Xi_2) = 51 + 41 = 92 > \Pi - \psi^* = 119 - 80 = 39$, $(A_0A_3A_2A_1)$ is not feasible due to its violation of the residency time constraint.

With $(A_0A_1A_3A_2)$, by (3.5), we have:
$\xi_0 = \alpha_0 + \alpha_1 + 6\lambda + 4\mu = 20 + 48 + 8 = 76$,
$\xi_1 = \alpha_1 + 8\lambda + 7\mu = 98 = \psi^*$,
$\xi_2 = \alpha_2 + \alpha_1 + 6\lambda + 4\mu = 106$, and
$\xi_3 = (\alpha_3 + 4\lambda + 3\mu)/2 = 119$.

Hence, $\Pi = \max\{\xi_0, \xi_1, \xi_2, \xi_3, \psi^*\} = 119$. By (3.6), we have:
$\Xi_1 = \xi_1 + \delta_1 = \xi_1 + 10 = 108 < \Pi = 119$,
$\Xi_2 = \xi_2 + 10 = 116 < \Pi = 119$, and
$\Xi_3 = \xi_3 + \delta_3/2 = 119 + 5 = 124 > \Pi = 119$.

Hence, we have $1, 2 \in \mathbf{E}$ and $3 \in \mathbf{F}$. Since $\partial = \sum_{j \in E} m_j \times (\Pi - \Xi_j) = (\Pi - \Xi_1) + (\Pi - \Xi_2) = 14 < \Pi - \psi^* = 119 - 98 = 21$, the conditions in Lemma 4.3 are satisfied. Then, according to Problem (4.2), we have linear program (4.4).

$$Min = \omega_0^*$$
$$\begin{cases} \omega_0^* + \omega_2^* + \omega_3^* \ge 11 \\ \omega_0^* + \omega_1^* \ge 3 \\ \omega_1^* + \omega_3^* \le 43 \\ \omega_0^* + \omega_2^* + \omega_3^* \le 21 \\ \omega_0^* + \omega_1^* \le 13 \\ \omega_2^* \le 0 \\ \omega_0^* + \omega_1^* + \omega_2^* + \omega_3^* = 21 \\ \omega_0^*, \omega_1^*, \omega_2^*, \omega_3^* \ge 0 \end{cases} \quad (4.4)$$

By solving (4.4) we have $\omega_0^* = \omega_2^* = 0$, $\omega_1^* = 10$, and $\omega_3^* = 11$. Then, by Lemma 4.2, we set $\omega_0 = \omega_0^* = 0$, $\omega_1 = \omega_1^* + \alpha_1 = 30$, $\omega_2 = \omega_2^* = 0$, and $\omega_3 = \omega_3^* = 11$ such that an optimal and feasible one-wafer cyclic schedule is found.

VI. Conclusions

With a backward strategy, the scheduling problem of single-robot-arm cluster tools with wafer residency time constraint is well-studied in the literature. However, it is more

practical and challenging to schedule one whose optimal schedule strategy is not limited to be backward. Up to now, this problem still remains open and this work aims to tackle it. To do so, the system is modeled by a Petri net. By following the existing result, it presents a method to find the optimal scheduling strategy for the system. Based on the Petri net model and obtained strategy, it is found that the key issue to schedule such a tool is to determine when and how long the robot should wait for. Based on this the schedulability is analyzed and necessary and sufficient conditions under which a feasible one-wafer cyclic schedule exists are developed. With the schedulability conditions, corresponding linear program is proposed to find an optimal and feasible schedule if existing. To find such a schedule, in a worst case, one needs to solve a linear program. Thus, it is computationally efficient.

Note that the proposed method provides a generalized framework for scheduling automated manufacturing systems with time window constraint, such as robot cells and hoist-based production systems. And we think that the obtained result can be extended to a single-arm multi-cluster tool. What we need to do are that: first, obtain the optimal scheduling strategy for each individual tool based on [Dawande *et al.*, 2002]; then, calculate the optimal cycle time of the system. Finally, with the optimal cycle time, one can derive the scheduling conditions in a similar way as Lemmas 4.1-4.3.

In this work, all the activity time is seen as constants. However, in practice, they may be subject to random variation such that the wafer sojourn time in a PM fluctuates, resulting in infeasibility. Furthermore, some wafer fabrication processes may need a wafer to visit some processing steps for more than once, leading to a revisiting process. In these cases, the scheduling problem of a cluster tool is much more challenging and is our future work.

REFERENCES

[1] M. Bader, R. Hall, and G. Strasser, Integrated processing equipment, *Solid State Technology.*, vol. 33, no. 5, pp. 149-154, 1990.

[2] P. Burggraaf, Coping with the high cost of wafer fabs, *Semi-conduct. Int*., vol. 38, pp.45-50, 1995.

[3] W. K. Chan, J. G. Yi, and S. W. Ding, "Optimal Scheduling of Multi-cluster Tools with Constant Robot Moving Times, Part I: Two-Cluster Analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 8, pp. 5-16, 2011.

[4] Y. Crama and J. van de Klundert, "Cyclic scheduling of identical parts in a robotic cell," *Operations Research*, vol. 45, no. 6, pp. 952-965, 1997.

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

[6] H. N. Geismar, C. Sriskandarajah, and N. Ramanan, Increasing throughput for robotic cells with parallel machines and multiple robots, *IEEE Transaction on Automation Science and Engineering*, vol. 1, no. 1, 84-89, 2004.

[7] H. Hu and M. C. Zhou, "A Petri Net-based Discrete Event Control of Automated Manufacturing Systems with Assembly Operations," IEEE Trans. on Control Systems Technology, 23(2), 513 - 524, Mar. 2015a.

[8] H. Hu, Y. Liu and M. C. Zhou, "Maximally Permissive Distributed Control of Large Scale Automated Manufacturing Systems Modeled with Petri Nets," IEEE Transactions on Control Systems Technology, 23(5), pp. 2026-2034, Sept. 2015b.

[9] J.-H. Kim, T.-E. Lee, H.-Y. Lee, and D.-B. Park, Scheduling analysis of timed-constrained dual-armed cluster tools, *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 3, 521-534, 2003.

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

[11] H.-Y. Lee and T.-E. Lee, "Scheduling single-arm cluster tools with reentrant wafer flows," *IEEE Transactions on Semiconductor Manufacturing*, vol.19, no. 2, 226-240, 2006.

[12] T. K. McNab, Cluster tools, part 1: emerging processes, *Semiconductor Int*., vol. 13, no. 9, 58-63, 1990.

[13] Newboe, Cluster tools: a process solution, *Semiconductor Int*., vol. 13, no. 8, 82-88, 1990.

[14] Y. Qiao, N. Q. Wu, and M. C. Zhou, Petri net modeling and wafer sojourn time analysis of single-arm cluster tools with residency time constraints and activity time variation, *IEEE Transactions on Semiconductor Manufacturing*, vol. 25, no. 3, 432-446, 2012.

[15] Y. Qiao, N. Q. Wu, and M. C. Zhou, Scheduling of dual-arm cluster tools with wafer revisiting and residency time constraints, *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, 286-300, 2014.

[16] 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. on Systems, Man, & Cybernetics*: *Systems*, 45(3), 472 – 484, Mar. 2015.

[17] 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.

[18] P. Singer, The driving forces in cluster tool development, *Semiconductor Int*., vol. 18, no. 8, 113-118, 1995.

[19] S. Sechi, C. Sriskandarajah, G. Sorger, J. Blazewicz, and W. Kubiak, "Sequencing of parts and robot moves in a robotic cell," *Int. J. Flexible Manuf. Syst*., vol. 4, no. 3-4, pp. 331-358, 1992.

[20] N. Q. Wu, "Necessary and Sufficient Conditions for Deadlock-free Operation in Flexible Manufacturing Systems Using a Colored Petri Net Model," *IEEE Transaction on Systems, Man, and Cybernetics, Part C*, vol. 29, no. 2, pp. 192-204, 1999.

[21] N. Q. Wu, C. B. Chu, F. Chu, and M. C. Zhou, "A Petri net method for schedulability and scheduling problems in single-arm cluster tools with wafer residency time constraints," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 2, pp. 224 - 237, 2008.

[22] N. Q. Wu, F. Chu, C. Chu, and M. Zhou, "Petri Net-Based Scheduling of Single-Arm Cluster Tools with Reentrant Atomic Layer Deposition Processes," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 42-55, 2011.

[23] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Petri net modeling and cycle time analysis of dual-arm cluster tools with wafer revisiting," *IEEE Transactions on Systems, Man, & Cybernetics*: Systems, vol. 43, no. 1, pp. 196-207, 2013.

[24] N. Q. Wu and M. C. Zhou, "Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems based on a Petri net model," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 5, 658-669, 2001.

[25] N. Q. Wu and M. C. Zhou, *System Modeling and Control with Resource-oriented Petri Nets*, CRC Press, Taylor & Francis Group, New York, October 2009.

[26] N. Q. Wu and M. C. Zhou, Process vs resource-oriented Petri net modeling of automated manufacturing systems, *Asian Journal of Control*, vol. 12, no. 3, 267-280, 2010a.

[27] N. Q. Wu and M. C. Zhou, "A closed-form solution for schedulability and optimal scheduling of dual-arm cluster tools based on steady schedule analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, 303-315, 2010b.

[28] N. Q. Wu and M. C. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 1, 203-209, 2012.

[29] M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri net approach*, World Scientific, Singapore, 1998.

Fajun Yang (S'16) received the B. S. degree in Industrial Engineering from Hunan University of Science and Technology, Hunan, China, in 2011, the Ph. D. degree in Mechanical Engineering from Guangdong University of Technology, China, in 2016. From 2015-2016, he was a Visiting Student with New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Research Fellow with Nanyang Technological University, Singapore. He has a number of journal and confrence papers. His interests are Petri nets, production planning, discrete event systems, scheduling and control. He has served as a reviewer for a number of journals.

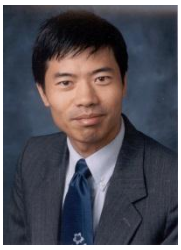
NaiQi Wu (M'04-SM'05) received his B. S. Degree in Electrical Engineering from Anhui University of Technology, Huainan, China, in 1982, the M. S. and Ph. D. Degrees in Systems Engineering both from Xi'an Jiaotong University, Xi'an, China in 1985 and 1988, respectively. From1988 to 1995, he was with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, and from 1995 to 1998, with Shantou University, Shantou, China. He moved to Guangdong University of Technology, Guangzhou, China in 1998. He joined Macau University of Science and Technology, Macau in 2013. He is currently a Professor at the Institute of Systems Engineering, Macau University of Science and Technology, Macau. His research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri net theory and applications, intelligent transportation systems, and energy systems. He is the author or coauthor of one book, five book chapters, and 130+ peer-reviewed journal papers. Dr. Wu was an associate editor of the IEEE Transactions on Systems, Man, & Cybernetics, Part C, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Systems, Man, & Cybernetics: Systems, and editor in chief of Industrial Engineering Journal.

Yan Qiao (M'16) received the B. S. and Ph. D. degrees in Industrial Engineering and Mechanical Engineering from the Guangdong University of Technology, Guangzhou, China, in 2009 and 2015, respectively.

From 2014 to 2015, he was a Visiting Student with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Post-Doctoral Research Associate with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau. He has one book chapter and 20+ international journal papers (majority in the IEEE Transactions). His research interests include discrete event systems, production planning, Petri nets, scheduling and control.

Dr. Qiao was a recipient of the QSI Best Application Paper Award Finalist of 2011 IEEE International Conference on Automation Science and Engineering, the Best Student Paper Award of 2012 IEEE International Conference on Networking, Sensing and Control, and the Best Conference Paper Award Finalist of 2016 IEEE International Conference on Automation Science and Engineering. He has served as a reviewer for a number of journals.

Rong Su (M'11-SM'14) received the B.E. degree in automatic control from the University of Science and Technology of China, Hefei, China, in 1997, and the M.A.Sc. and Ph.D. degrees both in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2000 and 2004, respectively. Since then he was affiliated with University of Waterloo and Technical University of Eindhoven before he joined Nanyang Technological University, Singapore, in 2010. His research interests include discrete event systems, supervisory control, model-based fault diagnosis, multiagent systems, optimization and scheduling with applications in green buildings, flexible manufacturing, power management, and intelligent transportation systems. In the aforementioned areas, he has more than 130 publications in journals, book chapters, and conference proceedings, and two patents. Dr. Su is an Associate Editor for Journal of Discrete Event Dynamic Systems: Theory and Applications, Journal of Control and Decision, and Transactions of the Institute of Measurement and Control, and the Chair of IEEE Control Systems Society Technical Committee on Smart Cities.



MengChu Zhou (S'88-M'90-SM'93-F'03) received his B.S. degree in Control Engineering from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree in Automatic Control from Beijing Institute of Technology, Beijing, China in 1986, and Ph. D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined New Jersey Institute of Technology (NJIT), Newark, NJ in 1990, and is now a Distinguished Professor of Electrical and Computer Engineering. His research interests are in Petri nets, intelligent automation, Internet of Things, big data, and intelligent transportation. He has over 690 publications including 12 books, 370+ journal papers (over 270 in IEEE transactions), and 28 book-chapters. He is the founding Editor of IEEE Press Book Series on Systems Science and Engineering. He is a recipient of Humboldt Research Award for US Senior Scientists, Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE Systems, Man and Cybernetics Society. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is a Fellow of International Federation of Automatic Control (IFAC) and American Association for the Advancement of Science (AAAS).