2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Optimal One-Wafer Cyclic Scheduling Analysis of Hybrid Multi-Cluster Tools With One-Space Buffering Module*

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

***Abstract*—A multi-cluster tool with both single and dual-arm individual cluster tools is called a hybrid multi-cluster tool. To operate it, one needs to coordinate different types of robots for accessing the shared buffering modules. Aiming at finding a one-wafer periodic schedule such that the lower bound of cycle time can be reached, this work studies its scheduling when its bottleneck individual tool is process-bound. With a timed Petri net model, the scheduling problem is reduced to determine the waiting time of robots. Then, the conditions are presented, under which a one-wafer periodic schedule exists such that the lower bound of cycle time can be reached. Based on them, an efficient algorithm is given to check whether such a one-wafer periodic schedule exists. If so, it is found via simple calculation. An example is given to show its effectiveness.**

I. Introduction

Cluster tools have been widely used in semiconductor manufacturing. Extensive studies have been done in modeling and performance evaluation in [8- 10, 13, 16, 19, 22-24]. They reveal that a cluster tool mainly operates under a steady state in two different regions: transport and process-bound one. The robot moving time from one process module (PM) to another can be treated as a constant and is much shorter than wafer processing time [6]. Hence, given constant robot task and PM processing time, a backward schedule is optimal for single-arm cluster tools [6, 7] and a swap strategy is effective for dual-arm cluster tools [10].

As a semiconductor manufacturing process becomes more and more complex, multi-cluster tools are invented to accommodate industrial needs [1]. They are composed of a number of single-cluster tools in a linear or treelike topology. A tool formed by $K \geq 2$ tools is called K-cluster tool. In it, two adjacent tools are linked by a buffering module (BM) with one or two spaces in a non-cyclic fashion [2, 3, 17]. It may be composed of both single and dual-arm cluster tools and thus called a hybrid K-cluster tool.

* This work was supported in part by the National Natural Science Foundation of China under Grants 60974098, 61034004, and 61374148.

F. J. Yang, and Y. Qiao are with the Department of Industrial Engineering, School of Electro-Mechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: fjyanggdut@163.com and yqiao@mail2.gdut.edu.cn).

N. Q. Wu is with the Faculty of Information Technology, Macau University of Science and Technology, Macau and also the Department of Industrial Engineering, School of Electro-Mechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: nqwu@must.edu.mo).

M. C. Zhou is with The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China and also the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982 USA (e-mail: zhou@njit.edu).

For K-cluster tools, Jevtic [5] proposes a heuristic rule to schedule it. Nevertheless, it is difficult to evaluate its performance. In [4], their scheduling problem is modeled by an event graph and network model. Then, an approach is proposed to find a periodic schedule. However, the method is based on simulation rather than analytical study [2]. With robot moving time ignored, a decomposition method is presented in [17]. By considering robot moving time, the work [2, 3] presents a polynomial algorithm to find an optimal multi-wafer cyclic schedule.

A cyclic schedule with the minimal cycle time is desired for operating a K-cluster tool. The cycle time of a K-cluster tool is closely related to the fundamental period (FP) of each individual cluster tool. Let FP_i be the FP of the i-th individual tool and $\Pi = \max\{FP_1, FP_2, \ldots, FP_K\}$ be the lower bound of cycle time of a hybrid K-cluster tool. Up to now, no study has answered under what condition a periodic schedule, especially a one-wafer cyclic schedule, can be found such that Π can be reached for a hybrid K-cluster tool. This work intends to address this important issue in both theory and practice.

This work focuses on the scheduling problem of a linear hybrid K-cluster tool in which single-cluster tools are linked by BMs that can hold one wafer at a time without any processing function. A Petri net (PN) model is developed to describe a hybrid K-cluster tool with special attention being paid to the buffering modules. Based on it, this work derives the conditions under which the lower bound of cycle time can be reached. It then gives an efficient algorithm to check whether there is a one-wafer cyclic schedule for a hybrid tool such that its lower bound of cycle time can be reached. If so, this work finds such a schedule by simple calculation.

The remainder of the paper is organized as follows. The next section develops the PN model for the system. Based on it, Section III analyzes the dynamic behavior of individual cluster tools. Section IV presents the conditions under which there is a one-wafer periodic schedule such that the lower bound of cycle time can be reached and a resulting scheduling algorithm. Section V uses an example to show it. Finally, Section VI concludes this paper.

II. Petri Net Modeling

Without loss of generality, we make the following assumptions as done in [2, 3]: 1) a PM can process one wafer at a time; 2) no parallel module, or each step configures only one PM for wafer processing; 3) the wafers processed have the same recipe and visit a PM no more than once; and 4) the task time of robots is deterministic. Also, it is assumed that, there is at least one PM (not including the BM) in each individual cluster tool except that the tail (define later) one has at least two PMs.

Let $\mathbf{N}_n = \{1, 2, \ldots, n\}$ and $\mathbf{\Omega}_n = \{0\} \cup \mathbf{N}_n$. For a hybrid K-cluster tool, let C_i and R_i, $i \in \mathbf{N}_K$, denote the i-th cluster tool and the robot in it, respectively, where C_1 is the "head" tool with two loadlocks in it and C_K is the "tail" one. The BM shared by C_i and C_{i+1} is seen as the outgoing module for C_i and the incoming one for C_{i+1}, respectively. Step j in C_i is denoted as PS_{ij}. Then, the outgoing BM for C_i is denoted as $PS_{i(b[i])}$, and the incoming BM for C_{i+1} is denoted as $PS_{(i+1)0}$. As the loadlocks for C_1 can be seen as an incoming module and it is denoted as PS_{10}. Let $n[i] + 1$ be the number of steps in C_i. Then, the steps in C_i are denoted as PS_{i0}, PS_{i1}, …, $PS_{i(b[i])}$, $PS_{i(b[i]+1)}$, …, $PS_{i(n[i])}$ with PS_{i0} and $PS_{i(b[i])}$ being the incoming and outgoing ones, respectively.

A. Petri Nets for Hybrid K-Cluster Tools

To model a linear hybrid K-cluster tool, we extend the resource-oriented PN developed in [11, 15]. Its basic concept is based on [18, 20, 21]. It is a kind of directed graphs containing places and transitions defined as PN = (P, T, I, O, M, K), where P is a finite set of places; T is a finite set of transitions with $P \cup T \neq \varnothing$ and $P \cap T = \varnothing$; $I : P \times T \to \mathbf{N} = \{0, 1, 2, \ldots\}$ is an input function; O: $P \times T \to \mathbf{N}$ is an output function; $M : P \to \mathbf{N}$ is a marking representing the number of tokens in places with M_0 being the initial marking; and K: $P \to \{1, 2, 3, \ldots\}$ is a capacity function where $K(p)$ represents the largest number of tokens that p can hold at any time.

The preset of transition t is the set of all input places to t, namely, ${}^\bullet t = \{p: p \in P \text{ and } I(p, t) > 0\}$. Its postset is the set of all output places from t, namely, $t^\bullet = \{p: p \in P \text{ and } O(p, t) > 0\}$. Similarly, ${}^\bullet p = \{t \in T: O(p, t) > 0\}$ and $p^\bullet = \{t \in T: I(p, t) > 0\}$. The transition enabling and firing rules can be found in [11, 15].

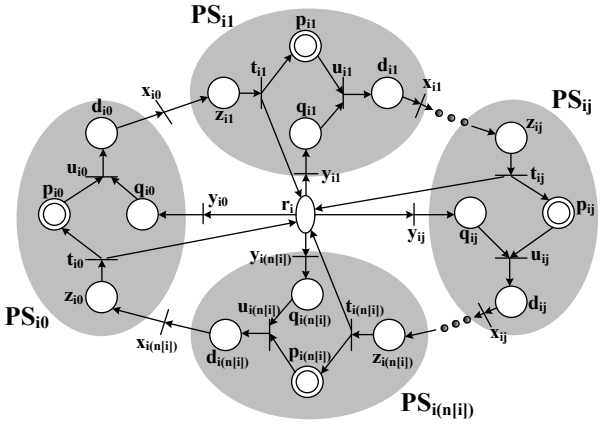

Fig. 2.1. The PN model for a single-arm cluster tool C_i, $i \in \mathbf{N}_K$.

We first present the PN model for a single-arm tool C_i, $i \in \mathbf{N}_K$. Step j in C_i, $j \in \mathbf{\Omega}_{n[i]}$, is modeled as follows. Timed place p_{ij} is used to model wafer processing Step PS_{ij} with $K(p_{ij}) = 1$. Timed place q_{ij} with $K(q_{ij}) = 1$ is introduced to model R_i's waiting before unloading a token from p_{ij}. Non-timed places z_{ij} and d_{ij} are added to model that R_i holds a token for loading into p_{ij} and moving to step $j + 1$ (or step 0 if $j = n[i]$), respectively. Timed transitions t_{ij} and u_{ij} are used to model that R_i loads a token into p_{ij} and unloads a token from p_{ij}, respectively. Then, by adding arcs (z_{ij}, t_{ij}), (t_{ij}, p_{ij}), (p_{ij}, u_{ij}), (q_{ij}, u_{ij}), and (u_{ij}, d_{ij}), the PN model for Step PS_{ij} in a single-arm tool C_i is obtained as shown in Fig. 2.1. With the PN model for a step, the wafer flow in C_i can be modeled as follows. Timed transition x_{ij} together with arcs (d_{ij}, x_{ij}) and $(x_{ij}, z_{i(j+1)})$ is used to model the event that R_i moves from Steps j to $j + 1$ (or Step 0 if $j = n[i]$, and arc $(x_{ij}, z_{i(j+1)})$ should be replaced by (x_{ij}, z_{i0})) with a wafer held. Robot R_i is modeled by place r_i with $K(r_i) = 1$, indicating that it has only one arm. A token in it represents that the arm is available. Timed transition y_{ij} together with arcs (r_i, y_{ij}) and (y_{ij}, q_{ij}) is used to model that R_i moves from Steps $j + 2$ to j (or from Step 0 if $j = n[i] - 1$, or from Step 1 if $j = n[i]$) without carrying a wafer. Finally, arc (t_{ij}, r_i) is added to represent that, by firing t_{ij}, the robot arm is released. In this way, the PN model for a single-arm tool C_i is obtained as shown in Fig. 2.1.

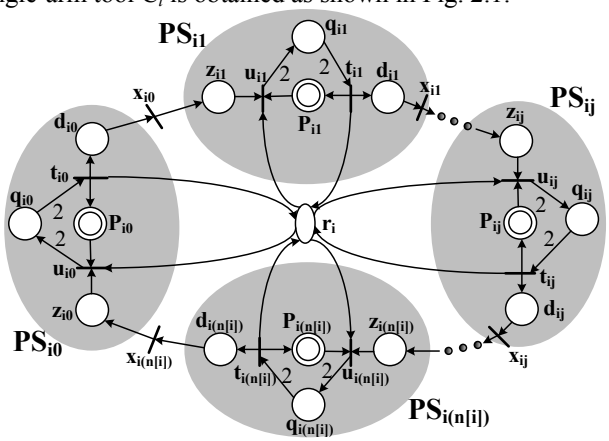

Fig. 2.2. The PN model for a dual-arm cluster tool C_i, $i \in \mathbf{N}_K$.

If C_i, $i \in \mathbf{N}_K$, is a dual-arm one, for Step j, $j \in \mathbf{\Omega}_{n[i]}$, timed place p_{ij} is used to model wafer processing at Step PS_{ij} with $K(p_{ij}) = 1$. Timed place z_{ij} with $K(z_{ij}) = 1$ models R_i's waiting before unloading a token from p_{ij}. As a swap strategy is adopted in a dual-arm tool, we use timed place q_{ij} with $K(q_{ij}) = 2$ to model the robot waiting during swap at p_{ij}. Non-timed place d_{ij} with $K(d_{ij}) = 1$ is used to model the state that a swap ends. Timed transitions u_{ij} and t_{ij} are used to model that R_i unloads a token from p_{ij} and loads a token into p_{ij}, respectively. Then, by adding arcs (z_{ij}, u_{ij}), (u_{ij}, q_{ij}), (q_{ij}, t_{ij}), (t_{ij}, d_{ij}), (t_{ij}, p_{ij}), and (p_{ij}, u_{ij}), the PN model for Step PS_{ij} is obtained as shown in Fig. 2.2. Then, we can obtain the PN model for a dual-arm tool in a similar way like single-arm tool as shown in Fig. 2.2.

For a BM that links C_i and C_{i+1}, $i \in \mathbf{N}_{K-1}$, there are four cases: 1) both C_i and C_{i+1} are dual-arm tools (D-D in short); 2) C_i is a single-arm tool and C_{i+1} is a dual-arm one (S-D); 3) C_i is a dual-arm tool and C_{i+1} is a single-arm one (D-S); and 4) both C_i and C_{i+1} are single-arm tools (S-S). Let the BM shared by C_i and C_{i+1} be $PM_{i(b[i])}$ as an outgoing module for C_i and $PM_{(i+1)0}$ as an incoming module for C_{i+1}. Places $p_{i(b[i])}$ and $p_{(i+1)0}$ with $p_{ib[i]} = p_{(i+1)0}$ and $K(p_{i(b[i])}) = K(p_{(i+1)0}) = 1$ are used to model it. The models of the four cases are shown in Fig. 2.3. Notice that $p_{i(b[i])}$ and $p_{(i+1)0}$ represent the same BM, therefore, when we refer to Step $b[i]$ in C_i we use $p_{i(b[i])}$, while to Step 0 in C_{i+1} we use $p_{(i+1)0}$.

With the PN structure developed, by putting a type of token W_0 representing a virtual wafer, the initial marking M_0 of the PN model is set as follows. First, set $M_0(p_{10}) = N$ to indicate that there are always wafers to be processed and the loadlocks can hold all these wafers.

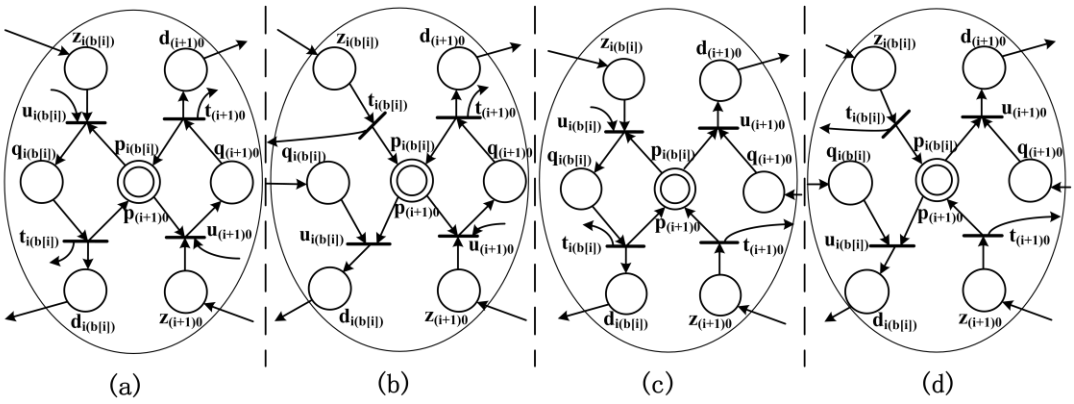

Fig. 2.3. The PN model for the buffering module between C_i and C_{i+1} that are (a) D-D, (b) S-D, (c) D- S, and (d) S-S cases.

If C_1 is a single-arm tool, we set $M_0(p_{1j}) = 1, j \in \mathbf{N}_{n[1]}$ and $j \neq b[1]$, $M_0(p_{1(b[1])}) = 0$; $M_0(z_{1j}) = M_0(d_{1j}) = M_0(r_1) = 0, j \in \mathbf{\Omega}_{n[1]}$; $M_0(q_{1j}) = 0, j \in \mathbf{\Omega}_{n[1]}$, and $j \neq b[1]$ - 1, and $M_0(q_{1(b[1]-1)}) = 1$, meaning that R_1 is waiting at Step $PS_{1(b[1]-1)}$ for unloading a wafer there. If C_1 is a dual-arm cluster tool, set $M_0(p_{1j}) = 1, j \in \mathbf{N}_{n[1]}$; $M_0(q_{1j}) = M_0(d_{1j}) = 0, j \in \mathbf{\Omega}_{n[1]}$; $M_0(r_1) = 1$ and $M_0(z_{1j}) = 0$, $j \in \mathbf{\Omega}_{n[1]}$, and $j \neq b[1]$, and $M_0(z_{1(b[1])}) = 1$, implying that R_1 is waiting at Step $PS_{1(b[1])}$ for unloading a wafer there.

For C_i, $i \in \mathbf{N}_K$ - {1}, if it is a single-arm tool, set $M_0(z_{ij}) = M_0(d_{ij}) = M_0(r_i) = 0, j \in \mathbf{\Omega}_{n[i]}$; $M_0(p_{ij}) = 1$ and $M_0(p_{i1}) = 0, j \in \mathbf{N}_{n[i]}$ - {1}; $M_0(q_{ij}) = 0, j \in \mathbf{N}_{n[i]}$, and $M_0(q_{i0}) = 1$, meaning that R_i is waiting at Step PS_{i0} for unloading a wafer there. If it is a dual-arm tool, set $M_0(p_{ij}) = 1, j \in \mathbf{N}_{n[i]}$; $M_0(q_{ij}) = M_0(d_{ij}) = 0, j \in \mathbf{\Omega}_{n[i]}$, $M_0(r_i) = 1$; $M_0(z_{ij}) = 0, j \in \mathbf{N}_{n[i]}$; and $M_0(z_{i0}) = 1$, implying that R_i is waiting at Step PS_{i0} for unloading a wafer there.

In the PN model for a BM, place $p_{i(b[i])}$ ($p_{(i+1)0}$) has two output transitions. For example, in Fig 2.3(c), both $u_{i(b[i])}$ and $u_{(i+1)0}$ are the output transitions of $p_{i(b[i])}$ ($p_{(i+1)0}$). To avoid conflict, colors are introduced into the model. We define the color for transitions and tokens as follows.

Definition 2.1: Define the color of a transition t_i as $C(t_i) = \{c_i\}$. If a token in $p \in {}^\bullet t_i$ enables t_i, then the token has the color of t_i, i.e., c_i.

By Definition 2.1, the colors of $u_{(i+1)0}$ and $u_{i(b[i])}$ in Fig 2.3 are $c_{(i+1)0}$ and $c_{i(b[i])}$, respectively. If a token enters $p_{i(b[i])}$ by firing $t_{i(b[i])}$, it has color $c_{(i+1)0}$ since it enables $u_{(i+1)0}$; while a token enters $p_{i(b[i])}$ by firing $t_{(i+1)0}$, it has color $c_{i(b[i])}$. In this way, the conflicts are completely resolved. However, the PN model is deadlock-prone. To solve this problem, we make the PN be a controlled PN. The control policy is defined as follows.

Definition 2.2: In a PN for single-arm tool C_i , at marking M, y_{ij}, $i \in \mathbf{N}_K$, $j \in \mathbf{\Omega}_{n[i]}$ - $\{n[i], b[i] - 1\}$, is control-enabled if $M(p_{i(j+1)}) = 0$; $y_{i(b[i]-1)}$ is control-enabled if transition $t_{i(b[i]+1)}$, $i \in \mathbf{N}_{K-1}$, has just been executed; $y_{i(n[i])}$ is control-enabled if transition t_{i1}, $i \in \mathbf{N}_K$, has just been executed.

B. Modeling Activity Time

In the developed PN model, both transitions and places represent activities that take time. If time ζ is associated with transition t, it means that firing t takes ζ time units. If it is associated with place p, a token in p must stay there for at least ζ time units before it can enable its output transition. The symbols and time durations for different places and transitions of C_i, $i \in \mathbf{N}_K$, are listed in Table I.

TABLE I. Time duration associated with transitions and places in C_i

Symbol	Transition or place	Action	Time	Tool type
λ_i	$t_{ij} \in T$	R_i loads/unloads a wafer into/from Step $j, j \in \mathbf{\Omega}_{n[i]}$	λ_i	Single-arm one
λ_i	u_{ij} and $t_{ij} \in T$	Swap operation at Step $j, j \in \mathbf{\Omega}_{n[i]}$	λ_i	dual-arm one
μ_i	$x_{ij} \in T$	R_i moves from a step to another with a wafer held	μ_i	Single and dual-arm ones
μ_i	$y_{ij} \in T$	R_i moves from a step to another without holding a wafer	μ_i	Single-arm one
α_{ij}	$p_{ij} \in P$	A wafer's processing in $p_{ij}, j \in \mathbf{\Omega}_{n[i]}$	α_{ij}	Single and dual-arm ones
τ_{ij}	$p_{ij} \in P$	A wafer's processing and waiting in $p_{ij}, j \in \mathbf{\Omega}_{n[i]}$	$\geq \alpha_{ij}$	Single and dual-arm ones
ω_{ij}	$q_{ij} \in P$	R_i waits before unloading a wafer from Step $j, j \in \mathbf{\Omega}_{n[i]}$	$[0, \infty)$	Single-arm one
ω_{ij}	$z_{ij} \in P$	R_i waits at $p_{ij}, j \in \mathbf{\Omega}_{n[i]}$	$[0,\infty)$	Dual-arm one
ω_{ij1}	$q_{ij} \in P$	R_i waits during swap at $p_{ij}, j \in \mathbf{\Omega}_{n[i]}$	0	Dual-arm one
	$d_{ij} \in P$	No robot activity is associated	0	Single and dual-arm ones
	$z_{ij} \in P$	No robot activity is associated	0	Single-arm one

III. Dynamic Behavior of Individual Cluster Tools

With the PN model, from [12], for single-arm tool C_i, $i \in \mathbf{N}_K$, the time taken for completing a wafer at step j in C_i is $\theta_{ij} = \alpha_{ij} + 4\lambda_i + 3\mu_i + \omega_{i(j-1)}, j \in \mathbf{N}_{n[i]}$, and $\theta_{i0} = \alpha_{i0} + 4\lambda_i + 3\mu_i + \omega_{i(n[i])} = 4\lambda_i + 3\mu_i + \omega_{i(n[i])}$. If R_i's waiting time is removed, the time taken for completing a wafer at Step j is

$$\xi_{ij} = \alpha_{ij} + 4\lambda_i + 3\mu_i, j \in \mathbf{N}_{n[i]} \quad (3.1)$$

and

$$\xi_{i0} = 4\lambda_i + 3\mu_i \quad (3.2)$$

With τ_{ij} being the wafer sojourn time in a PM, the cycle time at Step j in C_i is

$$\pi_{ij} = \tau_{ij} + 4\lambda_i + 3\mu_i + \omega_{i(j-1)}, j \in \mathbf{N}_{n[i]} \quad (3.3)$$

and $\pi_{i0} = \tau_{i0} + 4\lambda_i + 3\mu_i + \omega_{i(n[i])}$ (3.4)

The robot cycle time for a single-arm tool is

$$\psi_i = 2(n[i] + 1)(\lambda_i + \mu_i) + \sum_{j=0}^{n[i]} \omega_{ij} = \psi_{i1} + \psi_{i2} \quad (3.5)$$

where $\psi_{i1} = 2(n[i] + 1)(\lambda_i + \mu_i)$ is the robot cycle time without waiting and is a constant, $\psi_{i2} = \sum_{j=0}^{n[i]} \omega_{ij}$ is the robot waiting time in a cycle.

For dual-arm cluster tool C_i, $i \in \mathbf{N_K}$, according to [14], the time needed for completing a wafer at Step j in C_i is

$$\xi_{ij} = \alpha_{ij} + \lambda_i, j \in \mathbf{\Omega_{n[i]}} \quad (3.6)$$

The cycle time at Step j in C_i is

$$\pi_{ij} = \tau_{ij} + \lambda_i, j \in \mathbf{\Omega_{n[i]}} \quad (3.7)$$

The robot cycle time for dual-arm cluster tool C_i is

$$\psi_i = \sum_{j=0}^{n[i]} \omega_{ij} + (n[i] + 1)(\lambda_i + \mu_i) = \psi_{i2} + \psi_{i1} \quad (3.8)$$

where $\psi_{i2} = \sum_{j=0}^{n[i]} \omega_{ij}$ and $\psi_{i1} = (n[i] + 1)(\lambda_i + \mu_i)$ are the robot waiting and robot task time, respectively.

For each C_i, as the manufacturing process is serial. Therefore, in the steady state, for each step, the cycle time must be same and should be equal to the robot cycle time. Thus, C_i should be scheduled such that

$$\pi_i = \pi_{i0} = \pi_{i1} = \ldots = \pi_{i(n[i])} = \psi_i \quad (3.9)$$

IV. Scheduling Hybrid *K*-Cluster Tool

A. Schedule Properties

It follows from (3.9) that, to schedule an individual tool in a hybrid tool, the key is to determine its cycle time π_i. Let $\Pi_i = \max\{\xi_{i0}, \xi_{i1}, \ldots, \xi_{i(n[i])}, \psi_{i1}\}$ be the fundamental period (FP) of C_i, and $\Pi = \max\{\Pi_1, \Pi_2, \ldots, \Pi_K\}$. Assume that C_h, $1 \le h \le K$, is the bottleneck, or $\Pi = \Pi_h$. As discussed above, we assume that C_h is process-bound in this paper and such a tool is called process-dominant hybrid *K*-cluster tool. In the following discussion, because of the space limitation, we present the obtained results by omitting their proof. We have the following results.

Proposition 4.1: For a process-dominant hybrid *K*-cluster tool with C_h being the bottleneck, if tool C_i is scheduled such that its cycle time is π_i, $i \in \mathbf{N_K}$, further, if there exists a cyclic schedule with cycle time π, then,

$$\pi_i = \pi \ge \Pi, \forall i \in \mathbf{N_K} \quad (4.1)$$

It follows from Propositions 4.1 that we have the following result immediately for a one-wafer cyclic schedule such that the lower bound of cycle time is reached.

Theorem 4.1: To find a periodic schedule reaching the lower bound of cycle time, the cycle time of all individual tools in a process-dominant hybrid *K*-cluster tool is Π.

$$\pi_i = \pi = \Pi, \quad \forall i \in \mathbf{N_K} \quad (4.2)$$

By Theorem 4.1, it means that each individual cluster tool C_i in a hybrid *K*-cluster should be scheduled such that $\pi_i \ge \Pi$. In developing the scheduling method, we assume that after R_i loads a wafer into $p_{i(b[i])}$, $i \in \mathbf{N_{K-1}}$, R_{i+1} unloads the wafer from $p_{(i+1)0}$ ($p_{i(b[i])}$) immediately, for one can always schedule the system in this way. According to the scheduling method for an individual cluster tool, to optimally schedule a process-dominant hybrid *K*-cluster tool is to determine ω_{ij}'s for $i \in \mathbf{N_K}$ and $j \in \mathbf{\Omega_{n[i]}}$ such that the multiple robots can be optimally coordinated. If the system is scheduled such that the cycle time of each tool is Π and, at any marking M, a) when R_i, $i \in \mathbf{N_K}$, is scheduled to load (firing $t_{i(b[i])}$) a wafer into $p_{i(b[i])}$ ($p_{(i+1)0}$), $M(p_{(i+1)0}) = 0$ ($M(p_{i(b[i])}) = 0$) and, b) while R_i is scheduled to unload (firing $u_{i(b[i])}$) a wafer from $p_{(i+1)0}$ ($p_{i(b[i])}$), $M(p_{(i+1)0}) = 1$ ($M(p_{i(b[i])}) = 1$). Then, a one-wafer cyclic schedule with Π as its cycle time is obtained. Notice that there is no outgoing buffer in C_K, i.e., $p_{Kb[K]}$ does not exist, and for the loadlocks, $M(p_{10}) = N$ is always met. To obtain such a schedule, we should present condition(s) under which each BM that links C_i and C_{i+1}, $i \in \mathbf{N_{K-1}}$, is operated by coordinating R_i and R_{i+1} such that the cycle time of a hybrid *K*-cluster reaches its lower bound. We have the following results.

Lemma 4.1: For a process-dominant *K*-cluster tool, if the pair of C_i and C_{i+1}, $i \in \mathbf{N_{K-1}}$, is D-D or S-D case, there is always a cyclic schedule for both C_i and C_{i+1} such that their cycle time reaches the lower bound.

It follows from Lemmas 4.1 that for D-D and S-D, there is always a cyclic schedule for both C_i and C_{i+1} such that their cycle time reaches the lower bound. Thus, to know if a process-dominant hybrid *K*-cluster tool can reach the lower bound, we need examine D-S and S-S cases only. As for them, there is no guarantee that there are schedules for both C_i and C_{i+1} such that the lower bound of cycle time can be reached. Next, we identify the conditions under which both C_i and C_{i+1} can be scheduled to reach the lower bound of cycle time.

Theorem 4.2: For a process-dominant hybrid *K*-cluster tool, to obtain a one-wafer cyclic schedule such that the lower bound of cycle time is reached, if and only if, for C_i and C_{i+1}, $i \in \mathbf{N_{K-1}}$, the following conditions are satisfied by determining ω_{ij}'s and $\omega_{(i+1)l}$'s, $j \in \mathbf{\Omega_{n[i]}}$ and $l \in \mathbf{\Omega_{n[i+1]}}$.

$$\pi_{ij} = \pi_{(i+1)l} = \Pi, j \in \mathbf{\Omega_{n[i]}} \text{ and } l \in \mathbf{\Omega_{n[i+1]}} \quad (4.3)$$

If the pair of C_i and C_{i+1} is D-S case

$$\Pi - \lambda_i \ge 4\lambda_{i+1} + 3\mu_{i+1} + \omega_{(i+1)(n[i+1])} \quad (4.4)$$

If the pair of C_i and C_{i+1} is S-S case

$$\Pi - (4\lambda_i + 3\mu_i + \omega_{i(b[i]-1)}) \ge 4\lambda_{i+1} + 3\mu_{i+1} + \omega_{(i+1)(n[i+1])} \quad (4.5)$$

Theorem 4.2 presents the conditions under which a one-wafer cyclic schedule with its cycle time being the lower bound can be obtained. Next, we find such a schedule if there is one for a given hybrid *K*-cluster tool.

B. Scheduling Hybrid K-Cluster Tool

From Theorem 4.2, it is known that, for a pair of C_i and C_{i+1}, $i \in \mathbf{N_{K-1}}$, linked by a BM in a hybrid *K*-cluster tool, if, for any case, the corresponding conditions given in Theorem 4.2 are satisfied, the system can always be scheduled such that the lower-bound of cycle time is reached. Notice that the conditions given in Theorem 4.2 are functions of ω_{ij}'s for a single-arm tool only and it has nothing to do with a dual-arm cluster tool. Thus, the key in scheduling a hybrid tool is to schedule all the single-arm cluster tools by determining proper ω_{ij}'s. For dual-arm tool C_i, we can simply set $\omega_{i0} = \psi_{i2} = \Pi - \psi_{i1}$ and $\omega_{ij} = 0$, $i \in \mathbf{N_{n[i]}}$. However, if C_i is a single-arm tool, to make the conditions given in Theorem 4.2 satisfied if it is

possible, we need to determine ω_{ij}'s such that $\omega_{i(b[i]-1)}$ and $\omega_{i(n[i])}$ are as small as possible. Thus, if C_K is a single-arm tool, we set $\omega_{Kj} = \min\{\Pi - (4\lambda_K + 3\mu_K + \alpha_{K(j+1)}), \Pi - \psi_{K1} - \sum_{j=0}^{j-1} \omega_{Kj}\}, j \in \mathbf{\Omega}_{n[K]} - \{n[K]\}$, such that ω_{Kj} can be as large as possible. At last, $\omega_{K(n[K])} = \Pi - \psi_{K1} - \sum_{j=0}^{n[K]-1} \omega_{Kj}$. Then, we need to check whether $\Pi - \lambda_{K-1} < 4\lambda_K + 3\mu_K + \omega_{K(n[K])}$ holds or not if the pair of C_{K-1} and C_K is D-S case. If yes, the system cannot reach the lower bound of cycle time, otherwise, we set $\omega_{(K-1)j} = 0, j \in \mathbf{N}_{n[K-1]}$, and $\omega_{(K-1)0} = \psi_{(K-1)2} = \Pi - \psi_{(K-1)1}$. If the pair of C_{K-1} and C_K is S-S case, we need to check whether $\Pi - (4\lambda_{K-1} + 3\mu_{K-1} + \omega_{(K-1)(b[K-1]-1)}) < 4\lambda_K + 3\mu_K + \omega_{K(n[K])}$ holds or not. If yes, the system cannot reach the lower bound of cycle time, otherwise, we set $\omega_{(K-1)(b[K-1]-1)} = \min\{\Pi - (4\lambda_{(K-1)} + 3\mu_{(K-1)}) - (4\lambda_K + 3\mu_K + \omega_{K(n[K])}), \Pi - \psi_{(K-1)1}\}$ as large as possible, and set $\omega_{(K-1)j} = \min\{\Pi - (4\lambda_{K-1} + 3\mu_{K-1} + \alpha_{(K-1)(j+1)}), \Pi - \psi_{(K-1)1} - \omega_{(K-1)(b[K-1]-1)} - \sum_{j=0}^{j-1} \omega_{(K-1)j}\}$ as large as possible, $j \in \mathbf{\Omega}_{n[K-1]} - \{n[K-1], b[K-1]-1\}$, at last, $\omega_{(K-1)(n[K-1])} = \Pi - \sum_{j=0}^{n[K-1]-1} \omega_{(K-1)j} - \psi_{(K-1)1}$. Then, set the robot waiting time for C_{K-2}. By this way, we can determine the robot waiting time for C_{K-2}, C_{K-3}, …, and C_1 sequentially. In summary, we can schedule a process-dominant hybrid K-cluster tool by scheduling the individual cluster tool one by one as presented next.

Algorithm 4.1: Procedure for scheduling a process-dominant hybrid K-cluster tool

1. Calculate Π_i and ψ_{i1}, $i \in \mathbf{N}_K$, and let $\Pi = \max\{\Pi_1, \Pi_2, \ldots, \Pi_K\}$.
2. Determine $\omega_{Kj}, j \in \mathbf{\Omega}_{n[K]}$, as follows
   1) $i = K$, if C_i is a single-arm tool.
      a) For $j = 0, 1, \ldots n[i] - 1$ do
      b) $\omega_{ij} = \min\{\Pi - (4\lambda_i + 3\mu_i + \alpha_{i(j+1)}), \Pi - \psi_{i1} - \sum_{j=0}^{j-1} \omega_{ij}\}$;
      c) $\omega_{i(n[i])} = \Pi - \psi_{i1} - \sum_{j=0}^{n[i]-1} \omega_{ij}$, $i \leftarrow i - 1$;
      d) If C_i is a single-arm tool, go to Step 3, else, go to Step 4.
   2) $i = K$, if C_i is a dual-arm tool.
      a) For $j = 1, 2, \ldots n[i]$ do
      b) $\omega_{ij} = 0$, $\omega_{i0} = \psi_{i2} = \Pi - \psi_{i1}$, $i \leftarrow i - 1$;
      c) If C_i is a single-arm tool, go to Step 6, else, go to Step 5.
3. a) If $\Pi - (4\lambda_i + 3\mu_i) - (4\lambda_{i+1} + 3\mu_{i+1} + \omega_{(i+1)(n[i+1])}) < 0$, the system cannot reach the lower bound of cycle time, and go to Step 7;
   b) Otherwise, $\omega_{i(b[i]-1)} = \min\{\Pi - (4\lambda_i + 3\mu_i) - (4\lambda_{i+1} + 3\mu_{i+1} + \omega_{(i+1)(n[i+1])}), \Pi - \psi_{i1}\}$;
   c) For $j \in \mathbf{\Omega}_{n[i]} - \{n[i], b[i]-1\}$, do
   d) $\omega_{ij} = \min\{\Pi - (4\lambda_i + 3\mu_i + \alpha_{i(j+1)}), \Pi - \psi_{i1} - \omega_{i(b[i]-1)} - \sum_{j=0}^{j-1} \omega_{ij}\}$;
   e) $\omega_{i(n[i])} = \Pi - \psi_{i1} - \sum_{j=0}^{n[i]-1} \omega_{ij}$, $i \leftarrow i - 1$;
   f) If $i = 0$, go to Step vii. Otherwise, go to Step g;
   g) If C_i is a single-arm cluster tool, back to Step 3, else, go to Step 4.
4. a) If $\Pi - \lambda_i - (4\lambda_{i+1} + 3\mu_{i+1} + \omega_{(i+1)(n[i+1])}) < 0$, the system cannot reach the lower bound, and go to Step 7;
   b) Otherwise, for $j = 1, 2, \ldots, n[i]$ do
   c) $\omega_{ij} = 0$, $\omega_{i0} = \psi_{i2} = \Pi - \psi_{i1}$, $i \leftarrow i - 1$;
   d) If $i = 0$, then go to Step 7. Otherwise, go to Step e;
   e) If C_i is a single-arm tool, go to Step 6, else, go to Step 5.
5. a) For $j = 1, 2, \ldots, n[i]$ do
   b) $\omega_{ij} = 0$, $\omega_{i0} = \psi_{i2} = \Pi - \psi_{i1}$, $i \leftarrow i - 1$;
   c) If $i = 0$, then go to Step vii. Otherwise, go to Step d;
   d) If C_i is a single-arm tool, go to Step vi, else, back to Step v.
6. a) $\omega_{i(b[i]-1)} = \min\{\Pi - (4\lambda_i + 3\mu_i) - \lambda_{i+1}, \Pi - \psi_{i1}\}$;
   b) For $j \in \mathbf{\Omega}_{n[i]} - \{n[i], b[i]-1\}$, do
   c) $\omega_{ij} = \min\{\Pi - (4\lambda_i + 3\mu_i + \alpha_{i(j+1)}), \Pi - \psi_{i1} - \omega_{i(b[i]-1)} - \sum_{j=0}^{j-1} \omega_{ij}\}$;
   d) $\omega_{i(n[i])} = \Pi - \psi_{i1} - \sum_{j=0}^{n[i]-1} \omega_{ij}$, $i \leftarrow i - 1$;
   e) If $i = 0$, go to Step 7. Otherwise, go to Step f;
   f) If C_i is a single-arm tool, go back to Step 3, else, back to Step 4.
7. End

By Algorithm 4.1, for a hybrid K-cluster tool, we can check if there is a one-wafer cyclic schedule such that its lower-bound of cycle time is reached. This is stated as the following theorem.

Theorem 4.3: By Algorithm 4.1, a one-wafer cyclic schedule is obtained for a process-dominant hybrid K-cluster such that its cycle time reaches the lower-bound if it exists.

Theorem 4.4: The complexity of Algorithm 4.1 is constant.

V. Illustrative Example

For a hybrid 3-cluster tool, R_1 and R_2 are single-arm tools and R_3 is a dual-arm one. The activity time is as follows: for C_1, $\alpha_{10} = 0$ (the loadlocks), $\alpha_{11} = 40$s, $\alpha_{12} = 0$ (a BM), $\alpha_{13} = 41$s, $\alpha_{14} = 10$s, $\lambda_1 = 5$s, and $\mu_1 = 1$s; for C_2, $\alpha_{20} = 0$, $\alpha_{21} = 78$s, $\alpha_{22} = 0$, $\alpha_{23} = 60$s, $\alpha_{24} = 15$s, $\lambda_2 = 7$s, and $\mu_2 = 1$s; and for C_3, $\alpha_{30} = 0$, $\alpha_{31} = 64$s, $\alpha_{32} = 48$s, $\alpha_{33} = 85$s, $\alpha_{34} = 90$s, $\lambda_3 = 10$s, and $\mu_3 = 1$s.

For C_1, we have $\xi_{10} = \alpha_{10} + 4\lambda_1 + 3\mu_1 = 23$s, $\xi_{11} = 63$s, $\xi_{12} = 23$s, $\xi_{13} = 64$s, $\xi_{14} = 33$s, and $\psi_{11} = 2(n[1] + 1) \times (\lambda_1 + \mu_1) = 60$s. Hence, $\Pi_1 = 64$s. For C_2, we have $\xi_{20} = 31$s, $\xi_{21} = 109$s, $\xi_{22} = 31$s, $\xi_{23} = 91$s, $\xi_{24} = 46$s, and $\psi_{21} = 80$s. Hence, $\Pi_2 = 109$s, and C_2 is process-bound. For C_3, we have $\xi_{30} = 10$s, $\xi_{31} = 74$s, $\xi_{32} = 58$s, $\xi_{33} = 95$s, $\xi_{34} = 100$s, and $\psi_{31} = 55$s. Hence, $\Pi_3 = 100$s. Because $\Pi_2 = 109\text{s} > \Pi_3 = 100\text{s} > \Pi_1 = 64$s and C_2 is process-bound, this hybrid 3-cluster tool is process-dominant with $\Pi = 109$s. Then, according to the method presented in this paper, set $\pi_1 = \pi_2 = \pi_3 = \Pi = 109$s.

By Algorithm 4.1, with C_3 being a dual-arm one, set $\omega_{3j} = 0, j \in \mathbf{N}_{n[3]}$, and $\omega_{30} = \psi_{32} = \Pi - \psi_{31} = 109 - 55 = 54$s. For C_2, it

is a single-arm one and $\Pi - (4\lambda_2 + 3\mu_2) - \lambda_3 = 109 - 31 - 10 = 68 > 0$. Thus, according to Algorithm 4.1, we have $\omega_{2(b[2]-1)} = \omega_{21} = \min\{\Pi - (4\lambda_2 + 3\mu_2) - \lambda_3, \Pi - 2(n[2] + 1)(\mu_2 + \lambda_2)\} = 29$ and $\omega_{2j} = 0, j \in \mathbf{\Omega}_{n[2]}$ and $j \neq 1$. Then, for C_1, it is also a single-arm one and $\Pi - (4\lambda_1 + 3\mu_1) - (4\lambda_2 + 3\mu_2 + \omega_{2(n[2])} = 109 - 23 - 31 = 55 > 0$. Thus, by using Algorithm 4.1, we have $\omega_{1(b[1]-1)} = \omega_{11} = \min\{\Pi - (4\lambda_1 + 3\mu_1) - (4\lambda_2 + 3\mu_2 + \omega_{2(n[2])}, \Pi - 2(n[1] + 1)(\mu_1 + \lambda_1)\} = 49$ and $\omega_{1j} = 0, j \in \mathbf{\Omega}_{n[1]}$ and $j \neq 1$. In this way, a one-wafer cyclic schedule that reaches the lower bound of cycle time is obtained.

VI. Conclusion

Up to now, there is no research report on the optimal scheduling of process-dominant hybrid *K*-cluster tools. This paper for the first time presents the conditions under which a one-wafer cyclic schedule to reach its lower bound of cycle time exists. Furthermore, it gives an efficient algorithm to find such a schedule. With such a schedule, a hybrid *K*-cluster tool is clearly operated optimally, thereby guaranteeing the highest throughput. This work uses a PN model to describe the dynamic behavior of a hybrid *K*-cluster tool with special attention being paid to the buffering modules. It is our future work to obtain its minimal cycle time and optimal one-wafer cyclic schedule when a hybrid *K*-cluster tool cannot reach its lower bound.

References

[1] W. K. Chan, J. G. Yi, and S. W. Ding, "On the optimality of one-unit cycle scheduling of multi-cluster tools with single-blade robots," in *Proc. IEEE Int. Conf. Auto. Sci. Eng*., 392–397, Scottsdale, AZ, USA, 2007.

[2] 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.

[3] W. K. Chan, J. G. Yi, S. W. Ding, and D. Z. Song, "Optimal Scheduling of Multi-cluster Tools with Constant Robot Moving Times, Part II: Tree-Like Topology Configurations," *IEEE Trans. on Automation Science and Engineering*, vol. 8, pp. 17-28, 2011.

[4] S. W. Ding, J. G. Yi, and M. T. Zhang, "Multicluster Tools Scheduling: an Integrated Event Graph and Network Model Approach," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, pp. 339 - 351, 2006.

[5] D. Jevtic and S. Venkatesh, "Method and Apparatus for Scheduling Wafer Processing within a Multiple Chamber Semiconductor Wafer Processing Tool Having a Multiple Blade Robot," in U.S. Patent. vol. 6,224,638, 2001.

[6] 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.

[7] M.-J. Lopez and S.-C. Wood, Systems of multiple cluster tools - configuration, reliability, and performance, *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 2, 170-178, 2003.

[8] T. L. Perkinson, R. S. Gyurcsik, and P. K. MacLarty, "Single-wafer cluster tool performance: An analysis of the effects of redundant chambers and revisitations sequences on throughput," *IEEE Transactions on Semiconductor Manufacturing*, vol. 9, 384-400, 1996.

[9] T. L. Perkinson, P. K. MacLarty, R. S. Gyurcsik, and R. K. Cavin, III, "Single-wafer cluster tool performance: An analysis of throughput," *IEEE Trans. on Semiconductor Manufacturing*, vol. 7, 369-373, 1994.

[10] S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, A steady state throughput analysis of cluster tools: Dual-blade versus single-blade robots, *IEEE Trans. Semi-conduct. Manuf*., vol.10, pp. 418–424, 1997.

[11] N. Q. Wu, Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model, *IEEE Trans. On Systems, Man, and Cybernetics*, Part C, vol. 29, no. 2, 192-204, 1999.

[12] 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, pp. 224 - 237, 2008.

[13] 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, pp. 42-55, 2011.

[14] N. Q. Wu and M. Zhou, "Analysis of wafer sojourn time in dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Transactions on Semiconductor Manufacturing*, vol. 23, pp. 53-64, 2010.

[15] 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 Transactions on Robotics and Automation*, vol. 17, no. 5, 658-669, 2001.

[16] N. Q. Wu and M. C. Zhou, "Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 2, 446-454, 2012.

[17] J. G. Yi, S. W. Ding, D. Z. Song, and M. T. Zhang, "Steady-State Throughput and Scheduling Analysis of Multi-Cluster Tools for Semiconductor Manufacturing: A Decomposition Approach," *IEEE Trans. on Automation Science and Engineering*, vol. 5, pp. 321-336, 2008.

[18] M. C. Zhou and K. Venkatesh, *Modeling, simulation and control of flexible manufacturing systems: Petri net approach*, World Scientific, Singapore, 1998.

[19] W. M. Zuberek, "Timed Petri nets in modeling and analysis of cluster tools," *IEEE Transactions on Robotics Automation*, vol. 17, no. 5, pp. 562-575, Oct. 2001.

[20] S. G. Wang, C. Y. Wang, M. C. Zhou, and Z.W. Li, "A method to compute strict minimal siphons in S3PR based on loop resource subsets," *IEEE Trans. Systems*, *Man*, & *Cybernetics*, *Part A*, vol. 42, no. 1, 226–237, 2012.

[21] S. G. Wang, C. Y. Wang, and M. C. Zhou, "Design of Optimal Monitor-based Supervisors for a Class of Petri Nets with Uncontrollable Transitions," *IEEE Trans. Systems*, *Man*, & *Cybernetics*: *Systems*, vol. 43, no. 5, 1248–1255, 2013.

[22] Y. Qiao, N. Q. Wu, and M. C. Zhou, Real-Time Scheduling of Single-Arm Cluster Tools Subject to Residency Time Constraints and Bounded Activity Time Variation, *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 3, 564-577, 2012.

[23] 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 Constraint and Activity Time Variation, *IEEE Transactions on Semiconductor Manufacturing*, vol. 25, no. 3, 432-446, 2012.

[24] Q. Zhu and Y. Qiao, "Scheduling Single-Arm Multi-Cluster Tools with Lower Bound Cycle Time via Petri Nets," *Int. Journal of Intelligent and Control*, 17, pp. 113- 123, 2013.