



HAL
open science

A Petri-Net-Based Scheduling Strategy for Dual-Arm Cluster Tools With Wafer Revisiting

Naiqi Wu, Mengchu Zhou, Feng Chu, Chengbin Chu

► **To cite this version:**

Naiqi Wu, Mengchu Zhou, Feng Chu, Chengbin Chu. A Petri-Net-Based Scheduling Strategy for Dual-Arm Cluster Tools With Wafer Revisiting. *IEEE transactions on systems, man, and cybernetics*, 2013, 43 (5), pp.1182-1194. 10.1109/TSMCA.2012.2230440 . hal-00867371

HAL Id: hal-00867371

<https://hal.science/hal-00867371>

Submitted on 18 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Petri-Net-Based Scheduling Strategy for Dual-Arm Cluster Tools With Wafer Revisiting

NaiQi Wu, *Senior Member, IEEE*, MengChu Zhou, *Fellow, IEEE*,
Feng Chu, *Senior Member, IEEE*, and Chengbin Chu

Abstract—There are wafer fabrication processes in cluster tools that require wafer revisiting. The adoption of a swap strategy for such tools forms a 3-wafer cyclic (3-WC) period with three wafers completed in each period. It has been shown that, by such a scheduling strategy, the minimal cycle time cannot be reached for some cases. This raises a question of whether there is a scheduling method such that the performance can be improved. To answer this question, a dual-arm cluster tool with wafer revisiting is modeled by a Petri net. Based on the model, the dynamical behavior of the process is analyzed. Then, a 2-wafer cyclic (2-WC) scheduling strategy is revealed for the first time. Cycle time analysis is conducted for the proposed strategy to evaluate its performance. It shows that, for some cases, the performance obtained by a 2-WC schedule is better than that obtained by any existing 3-WC ones. Thus, they can be used to complement each other in scheduling dual-arm cluster tools with wafer revisiting. Illustrative examples are given.

Index Terms—Cluster tool, Petri nets (PNs), scheduling, semiconductor manufacturing.

I. INTRODUCTION

A CLUSTER tool is a kind of integrated, automated, and configurable equipment that adopts the advanced single-wafer processing technology in semiconductor manufacturing [1], [2]. It results in higher yield [10], shorter cycle time [8], [10], [15], better utilization of space [2], [15], and lower capital cost [15]. Thus, to gain higher profit, semiconductor manufacturers increasingly adopt cluster tools for their wafer fabrication processes. A cluster tool consists of a number of process modules (PMs), an aligner module, a wafer handling robot, and loadlocks for wafer cassette loading/unloading, which are

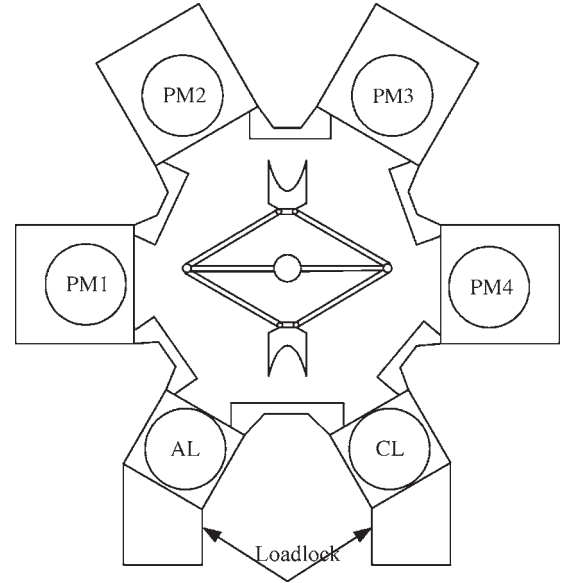


Fig. 1. Dual-arm cluster tool with four PMs.

mechanically linked together in a radial way. The robot can be a single- or a dual-arm one, and the corresponding cluster tools are called single- and dual-arm cluster tools, respectively. A dual-arm cluster tool is illustrated in Fig. 1. For a dual-arm robot, the two arms are tightly coupled by construction. Thus, at any time, only one of the two arms can pick up or drop off a wafer from or to a PM/loadlock.

In order to operate cluster tools effectively, great attention has been paid to their modeling, analysis, performance evaluation, and scheduling for cluster tools, such as the studies in [3], [11]–[13], [16], [28], [31]–[33], and [35]. With these studies, it is found that a cluster tool can operate under the steady state in one of the two regions, namely, transport-bound and process-bound ones. In the former, the robot is the critical resource such that it is always busy and the system cycle time is determined by robot task times. In the latter, it is determined by the processing times in the PMs, whereas the robot could have idle times. With these findings, a periodic schedule can be found for cluster tools. For single-arm cluster tools, the backward scheduling strategy is widely adopted and is shown to be optimal [6], [7]. For dual-arm cluster tools, with two arms available, a swap strategy can be applied and has been shown to be efficient and optimal [16], [31], [32]. It simplifies robot task scheduling and reduces robot cycle time. As pointed out in [14], in cluster tools, PM activities follow the robot tasks. Hence, it is crucial to schedule the robot tasks in scheduling cluster tools.

N. Wu is with the Department of Industrial Engineering, School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: nqwu@gdut.edu.cn).

M. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA, and also with Tongji University, Shanghai 200092, China (e-mail: zhou@njit.edu).

F. Chu is with the Laboratoire Informatique, Biologie Intégrative et Systèmes Complexes, Université d'Evry Val d'Essonne, 91020 Evry Cédex, France (e-mail: feng.chu@ibisc.univ-evry.fr).

C. Chu is with the Laboratoire Génie Industriel, Ecole Centrale Paris, 92295 Châtenay-Malabry Cedex, France, and also with Tongji University, Shanghai 200092, China (e-mail: chengbin.chu@cep.fr).

Some wafer fabrication processes, such as an atomic layer deposition (ALD) process, are repeated processes. Such a process requires a wafer visits some PMs several times, leading to a wafer revisiting process [5]. As pointed out in [5], often, the repeated process steps require exactly identical processing conditions when the wafers visit the same PM. It is shown that, for single-arm cluster tools with wafer revisiting, the process is deadlock prone [5]. Thus, it is necessary to schedule single-arm cluster tools such that they are deadlock free. Furthermore, with wafer revisiting, the backward scheduling is no longer optimal, and the process is not a flow shop such that the backward scheduling is not feasible. Therefore, it is much more complicated to schedule single-arm cluster tools with wafer revisiting than those without wafer revisiting. Petri net (PN) models are developed for their analysis and performance evaluation in [29] and [35]. The obtained results in these studies can be used as a scheduling guide, but they do not present any method to avoid deadlock and find an optimal schedule. To find a feasible and optimal schedule, Lee and Lee [5] presented a PN model to describe the dynamical behavior. Based on it, the scheduling problem is formulated as mixed integer programming. It can be solved in an exact way to find a deadlock-free and optimal schedule. Nevertheless, the approach involves prohibitive computation because its computational complexity is exponential with the number of discrete variables. A computationally efficient method is presented in [18] for single-arm cluster tools with wafer revisiting by using a PN model.

For dual-arm cluster tools with wafer revisiting, swap strategy scheduling can still be applied without any problem. With such a strategy adopted, PN models are developed to describe their behavior in [30] and [36]. By these models, when a swap strategy is applied, performance is analyzed, and it believes that a dual-arm cluster tool with wafer revisiting can reach its steady state. Then, based on the models, cycle time can be calculated, and it is shown that a swap strategy is optimal in the sense of cycle time. However, this result is shown to be not true in a recent study in [19] and [20] for the ALD process, which is a typical wafer revisiting process in wafer fabrication. It is found that, by applying a swap strategy, the process contains different cycles. If a cycle involves all processing steps of the system, it is called a global cycle. However, if it involves the revisiting process only, it is called a local cycle. With a swap strategy, the resulting schedule is still a periodic one, and a period in the process contains three local cycles in the revisiting process and three global cycles. Thus, the system repeatedly switches from the local cycles to the global cycles and from the global cycles to the local cycles. Consequently, the system may never reach its steady state; instead, it is always in a transient state. Furthermore, for some cases, the cycle time cannot reach its lower bound, i.e., optimal. Notice that, by a swap strategy, three wafers are completed in a period, and it is called a 3-wafer cyclic (3-WC) schedule. With this finding, it gives rise to a question of whether the 3-WC schedule is optimal or whether there exists a scheduling method that can improve the cycle time. This motivates us to conduct this study. In this paper, a PN model is developed to well describe the dynamical behavior of dual-arm cluster tools with wafer revisiting. With the model, by

analyzing the dynamical behavior of the process, it finds that, to improve the scheduling performance for such a system is to reduce the number of local and global cycles in a scheduling period. With this finding, a 2-wafer cyclic (2-WC) scheduling strategy that contains two local cycles and two global cycles is presented. Based on the proposed scheduling strategy, cycle time analysis is carried out to evaluate its performance. It shows that, for some situations, it can reach the lower bound of cycle time, whereas the 3-WC schedule cannot. This shows that the 2-WC scheduling outperforms the 3-WC scheduling for these situations.

The difference between this paper and our previous studies [19], [20] is as follows. As a swap strategy is widely used for scheduling dual-arm cluster tools, it is commonly believed that, when this strategy is applied to schedule dual-arm cluster tools with wafer revisiting, the process can enter its steady state, and it is optimal in the sense of cycle time. This result is shown to be incorrect in [19] and [20]. Based on this finding, analytical expressions are presented for the correct calculation of cycle time, but it presents no methods to improve the performance. With a PN model developed, by analyzing the dynamical behavior of the system, this paper presents new scheduling methods for such tools. It also presents how to calculate the cycle time for the new methods. Results show that the proposed scheduling methods are superior to 3-WC scheduling ones.

In Section II, a PN model is developed for dual-arm cluster tools with revisiting. Based on the analysis of dynamical behavior with the model, a 2-WC scheduling strategy is presented in Section III. Then, Section IV conducts the cycle time analysis under such a strategy. Several examples are used to show the results in Section V. Finally, Section VI gives the conclusions.

II. SYSTEM MODELING BY PN

Here, we first present the wafer revisiting patterns for a dual-arm cluster tool and then develop the PN model by extending our previous work [17], [21]–[27], [29]. Previously, a resource-oriented PN (ROPN) modeling method is proposed for manufacturing systems for deadlock avoidance. Based on the ROPN idea, this paper develops a PN model for the scheduling of dual-arm cluster tools with wafer revisiting.

A. Wafer Flow Patterns

The fabrication of a wafer in a cluster tool contains a number of operations. With wafer revisiting, some operations require the same fabrication process to be performed by a given set of PMs. Such a set of PMs is referred to as a processing step. Let PMD_i be the set of PMs for Step i and $|PMD_i|$ be the number of PMs in PMD_i . In [5], a typical and practical wafer revisiting process for ALD is presented, where a wafer is required to visit two consecutive processing steps, for example, Steps i and $i + 1$, for several times, as shown in Fig. 2. In order to balance the workload, more than one PM can be configured for performing the process in a step, and let $m_i = |PMD_i|$ be the number of PMs in Step i . If a wafer needs to visit some steps k times, this process is called a k -time revisiting pattern. A k -time revisiting process and the entire wafer flow pattern can be denoted by $(m_i, m_{i+1})^k$ and $(m_1, \dots, (m_j, m_{j+1})^k, \dots, m_n)$,

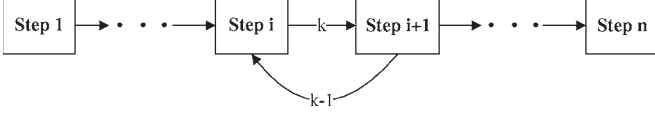


Fig. 2. Illustration of a wafer flow pattern with revisiting ($k > 1$).

respectively. In an ALD process, the wafer flow pattern is $(m_1, (m_2, m_3)^k)$ and, often, $m_1 = m_2 = m_3 = 1$. Since ALD is a typical and commonly used wafer fabrication process, this study is conducted based on this process.

A revisiting process may contain only one step or more than two steps. Notice that, as a typical revisiting process, the ALD process has a revisiting process with two steps. Moreover, the results obtained from a two-step revisiting process are useful for the other cases. Thus, for the sake of simplicity, only a two-step revisiting process is considered in this paper. A wafer fabrication process with revisiting in cluster tools may contain more than one revisiting process. However, the analysis of such a process can be done based on the process with only one revisiting process. Therefore, we address the wafer fabrication process with revisiting as shown in Fig. 2.

B. Finite Capacity PN

To schedule a cluster tool is to dynamically allocate the finite resources (robot arms and PMs) to the manufacturing tasks. PNs are powerful in modeling the behavior of resource allocation. With limited resources in a manufacturing system, a finite capacity PN is an ideal choice to model them. The concept of a PN presented here is based on [9], [27], [34], [37]–[40]. A finite capacity PN is denoted by $PN = (P, T, I, O, M_0, K)$, where $P = \{p_1, p_2, \dots, p_m\}$ is a finite set of places; $T = \{t_1, t_2, \dots, t_n\}$ is a finite set of transitions, $P \cup T \neq \emptyset$, and $P \cap T = \emptyset$; $I : P \times T \rightarrow N$ is an input function, where $N = \{0, 1, 2, \dots\}$; $O : P \times T \rightarrow N$ is an output function; $M : P \rightarrow N$ is a marking representing the numbers of tokens in places with M_0 denoting the initial marking; and $K : P \rightarrow \{1, 2, 3, \dots\}$ is a capacity function, where $K(p)$ represents the number of tokens that place p can hold at most at a time.

The *preset* of transition t is the set of all input places to t , i.e., $\bullet t = \{p : p \in P \text{ and } I(p, t) > 0\}$. The *postset* of t is the set of all output places from t , i.e., $t\bullet = \{p : p \in P \text{ and } O(p, t) > 0\}$. Similarly, p 's *preset* $\bullet p = \{t \in T : O(p, t) > 0\}$, and *postset* $p\bullet = \{t \in T : I(p, t) > 0\}$.

Definition 2.1: A transition $t \in T$ in a PN is enabled if $\forall p \in P$

$$M(p) \geq I(p, t) \quad (2.1)$$

$$K(p) \geq M(p) - I(p, t) + O(p, t). \quad (2.2)$$

When a transition is enabled, it can fire. Firing an enabled transition t at marking M yields, $\forall p \in P$

$$M'(p) = M(p) - I(p, t) + O(p, t). \quad (2.3)$$

Definition 2.1 indicates that t is enabled and can fire if all the places in $\bullet t$ have enough tokens and if all the places in $t\bullet$ have enough free spaces. Thereafter, when condition (2.1) is met, t is said to be process-enabled. When condition (2.2) is

met, t is said to be resource-enabled. Thus, t is enabled only if it is both process-enabled and resource-enabled. A sequence of firings yields a sequence of markings. In a PN, M_i is said to be reachable from M_0 if there exists a sequence of firings that transforms M_0 to M_i . The set of all markings reachable from M_0 is denoted by $R(M_0)$. A transition in a PN is live if it can fire at least once in some firing sequences for every marking M reachable from M_0 . A PN is live if every transition is live. The liveness of a PN guarantees that all events or activities modeled by the PN can happen. It implies the deadlock-free operation of an automated manufacturing system.

Time can be described in a PN by associating it with transitions and/or places. When time τ is associated with a transition, its firing should take τ time units. When it is associated with a place, a token should stay in it for at least τ time units before the token can enable its output transition(s).

C. Modeling Robot Activities

As pointed out in [14], PM activities in cluster tools follow the robot tasks, and thus, it is crucial to schedule robot activities. From the viewpoint of robot activities, a loadlock can be seen just as a PM. Therefore, the loadlocks can be treated just as a processing step when modeling. We let Step zero represent the loadlocks. Hence, the robot activities include unloading a wafer from a PM, moving from a PM to another with a wafer carried, waiting if the module is busy, loading a wafer into a PM, moving from a PM to another without carrying a wafer, and waiting. In a cluster tool, for a type of wafers, the flow pattern is deterministic. Hence, when a robot arm unloads a wafer from a PM, it must move to a prespecified destination PM and loads this wafer into that PM. Thus, these three activities should be modeled together as a macro activity. We use a macro transition t_{ij} to model this macro activity that consists of three transitions x_{ij} , y_{ij} , and z_{ij} ; and two places a_{ij} and b_{ij} with $K(a_{ij}) = K(b_{ij}) = 1$, as shown in Fig. 3. Transition x_{ij} is for the robot unloading a wafer from a PM at Step i , y_{ij} is for moving from Step i to Step j with a wafer held, and z_{ij} is for loading the wafer into a PM at Step j . When a robot arm moves to Step j with a wafer carried, it is possible that the wafer cannot be loaded into Step j due to the absence of any free space. Thus, it is necessary for the robot to wait before the wafer can be loaded. A token in place b_{ij} represents a robot arm waiting with a wafer held.

In the model, places r_1 and r_2 are used to model the availability of the robot arms with $K(r_1) = K(r_2) = 1$. A token in r_1 or r_2 means that one of the two arms is available, whereas a token in both r_1 and r_2 means that both arms are idle. In a dual-arm cluster tool, the two robot arms cannot perform activities simultaneously, which should be guaranteed by the model. We use transition t_0 to connect r_1 and r_2 such that they can act only in a serial way. By such modeling, when $M(r_1) = M(r_2) = 1$, or both arms are available, only the arm represented by the token in r_2 can be used to enable a transition. After the token in r_2 is removed by firing one of its output transitions, t_0 fires, and the token in r_1 goes into r_2 . Then, the other arm can act. This way, it correctly describes the detail of robot activities, and at the same time, it reflects the characteristic of the process.

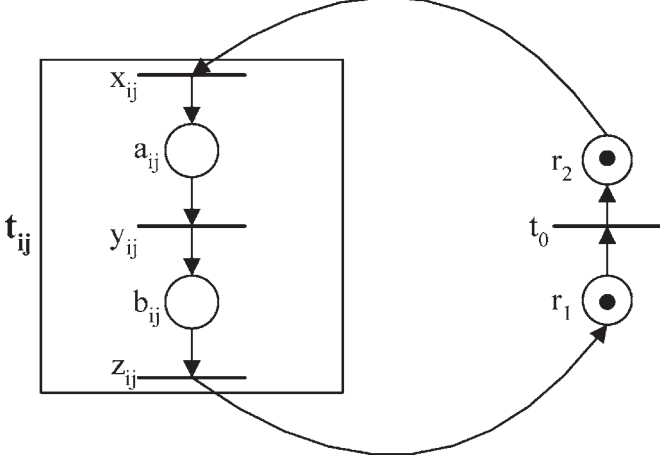


Fig. 3. PN model for robot activities.

D. Modeling the Wafer Flow

With the PN model for robot activities, we can model the wafer fabrication process as follows. We use place p_i with $K(p_i) = |\text{PMD}_i| = m_i$ to model the set of PMs in PMD_i for Step i . A token in p_i represents that a wafer is being processed by a PM in PMD_i . By $K(p_i) = m_i$, it means that p_i can hold, at most, m_i tokens at a marking. With two loadlocks available, when the wafers in one loadlock are being processed, another cassette of wafers can be loaded into the other loadlock. Hence, there are always wafers to be processed. The loadlocks are thought to be Step zero and are modeled by place p_0 with $K(p_0) = \infty$. Then, the wafer flow can be modeled as follows. Macro transitions t_{01} (with one arc from p_0 to t_{01} and another from t_{01} to p_1) and t_{n0} (with one arc from p_n to t_{n0} and another from t_{n0} to p_0) are used to connect p_0 and p_1 and p_n and p_0 , respectively. We put macro transition t_{ij} (with one arc from p_i to t_{ij} and another from t_{ij} to p_j) between p_i and p_j to connect p_i and p_j , if the wafers go to p_j after they are processed by p_i . For revisiting process $(m_i, m_{i+1})^k$, macro transition $t_{(i+1)i}$ is used to connect p_{i+1} and p_i . For example, if there is a three-step process with Steps 2 and 3 being the revisiting process, five macro transitions $t_{01}, t_{12}, t_{23}, t_{30}$, and t_{32} should be added, where t_{32} is for the revisiting process.

Then, it needs to model the movement of a free robot arm. We use transition h_{ij} to model moving a free robot arm to Step i for the purpose of delivering a wafer from Step i to Step j . At times, when t_{ij} fires such that a token is loaded into p_j , the robot arm needs to wait at p_j for the completion of the wafer represented by the token. In this case, we think that there is a virtual transition h_{jj} . Similarly, a free robot arm may need to wait for the completion of a wafer that is to be delivered from Step i to Step j . Place q_{ij} is used to model this waiting process with $K(q_{ij}) = 1$. Then, to complete the modeling of the wafer flow, we add an arc from h_{ij} to q_{ij} and an arc from q_{ij} to t_{ij} , $\forall h_{ij}, q_{ij}$, and t_{ij} ; an arc from t_{ij} to r_1 , $\forall t_{ij}$; and an arc from r_2 to h_{ij} , $\forall h_{ij}$. This way, a PN model for dual-arm cluster tools with wafer revisiting is developed. The PN model for wafer flow pattern $(m_1, (m_2, m_3)^k)$ is shown in Fig. 4.

Observing the PN shown in Fig. 4, it can be found that there are two simple circuits that are related to wafer flow, namely,

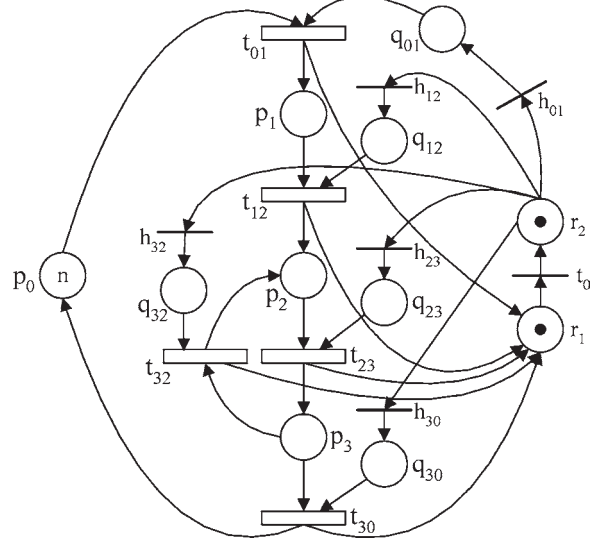


Fig. 4. PN model for wafer flow pattern $(m_1, (m_2, m_3)^k)$.

$(p_2, t_{23}, p_3, t_{32}, p_2)$ and $(p_0, t_{01}, p_1, t_{12}, p_2, t_{23}, p_3, t_{30}, p_0)$. The latter is not related to the revisiting process, whereas the former describes the revisiting process. In cluster tools with wafer revisiting, the number of revisiting times can be up to more than five times [5]. Hence, it is important for a PN to model such a process. In the PN shown in Fig. 4, macro transition t_{32} can fire as many times as required, or a multiple revisiting process is well modeled.

In the PN shown in Fig. 4, a token in p_3 may go through t_{30} or t_{32} . It needs to make sure which way a token in p_3 should go. This problem is solved by introducing colors into the PN. To do so, we define a unique color to a macro transition in the PN.

Definition 2.2: Define the unique color of a transition t_{ij} in the PN as $C(t_{ij}) = c_{ij}$.

With Definition 2.2, we can then define the colors for tokens. It is known that a token in r_1 or q_{ij} enables its unique output transition, or there is no confusion at all. A token in r_2 means that a robot arm is free to be used to unload a wafer from any step. It leaves the system controller to decide which step the robot arm should move to. This reflects the flexibility of scheduling a cluster tool. In other words, a token in r_2 enables every h_{ij} , and it is not necessary to give a color to this token. Thus, we need to define colors for tokens in p_i only. Based on the colors for transitions in the PN, we can define colors for tokens in p_i . Let W_d denote a token representing the d -th wafer released into the system.

Definition 2.3: Define the color of a token W_d in p_i as $C(p_i, W_d) = c_{ij}$ if the token goes to p_j after being processed in a PM of PMD_i modeled by p_i .

Let $W_d(f)$ in p_i be a token representing that the d -th wafer visits p_i for the f -th operation. Then, if the wafers need to visit the revisiting steps k times, we have $C(p_3, W_d(f)) = C(t_{32}) = c_{32}$, $1 \leq f < 2k + 1$, and $C(p_3, W_d(2k + 1)) = C(t_{30}) = c_{30}$. When a token W_d in p_i has color c_{ij} , it is said that W_d enables t_{ij} . For example, token $W_d(g)$, $1 \leq g < 2k + 1$, in p_3 enables t_{32} , whereas $W_d(2k + 1)$ in p_3 enables t_{30} . We use $M(p_i)$ to denote the number of tokens in p_i , regardless of the token color at marking M , and $M(p_i, c_{ij})$ to denote the

number of tokens in p_i at M that have color c_{ij} . Then, we can define the enabling and firing rules for the PN as in the succeeding discussion.

Definition 2.4: A macro transition t_{ij} , $i \neq 0$, in the PN is said to be enabled at M if

$$M(p_i, c_{ij}) \geq 1 \quad (2.4)$$

$$M(q_{ij}) = 1. \quad (2.5)$$

Observing Fig. 4, we know that, to fire a macro transition t_{ij} , there must be a token in both p_i and q_{ij} . A token in p_i represents that there is a wafer at Step i , whereas a token in q_{ij} represents that a free robot arm is ready to execute the activities of t_{ij} . This means that the key issue is to schedule the robot to determine which action should be executed next, as pointed out in [14]. To make a token enter q_{ij} , h_{ij} must fire. Thus, it is h_{ij} 's firing that schedules the robot and determines the evolution of the PN. Thus, it is very important to define the enabling rule for h_{ij} 's. Note that t_{ki} is a macro transition that contains two places, in which there can be tokens. Thus, we let $M(t_{ki}) = M(a_{ki}) + M(b_{ki})$ denote the number of tokens being used for firing t_{ki} at M . When $M(t_{ki}) = 1$, it means that the robot holds a wafer and waits to drop the wafer into p_i .

Definition 2.5: Transition h_{ij} with $t_{ki} \in \bullet p_i$ and $t_{ij} \in p_i \bullet$ in the PN is said to be enabled at M if one of the following conditions is satisfied:

$$1) \quad M(r_1) = M(r_2) = 1 \quad (2.6)$$

$$2) \quad \begin{aligned} M(p_i, c_{ij}) &\geq 1 \\ M(p_j) &< K(p_j) \\ M(r_1) + M(r_2) &= 1 \end{aligned} \quad (2.7)$$

$$3) \quad \begin{aligned} M(t_{ki}) &= 1 \\ M(p_i, c_{ij}) &\geq 1 \\ M(p_i) &= K(p_i) \\ M(r_1) + M(r_2) &= 1. \end{aligned} \quad (2.8)$$

Notice that using a robot arm to fire h_{ij} implies that this arm is used to fire macro transition t_{ij} . Thus, Condition 1) implies that, when both robot arms are free, any h_{ij} (or t_{ij}) can fire. This flexibility allows us to obtain a different scheduling strategy other than the swap one. Condition 2) guarantees that, when only one robot arm is free, this arm can be used to fire a macro transition t_{ij} whose output place p_j has at least one free space. Firing $t_{ki} \in \bullet p_i$ and $t_{ij} \in p_i \bullet$ forms a swap operation at p_i . Notice that, by $M(r_1) + M(r_2) = 1$, it means that there is only one robot arm available, and it does not imply that h_{ij} is enabled, even if $M(r_2) = 0$. In fact, if $M(r_1) = 1$ and $M(r_2) = 0$, t_0 immediately fires such that $M(r_1) = 0$ and $M(r_2) = 1$. Condition 3) says that, when a swap operation is initiated by firing t_{ki} with one robot arm, the other free robot arm should be used to complete the initiated swap operation by firing t_{ij} .

An enabled macro transition t_{ij} can then fire with firing x_{ij} , y_{ij} , and z_{ij} in a serial way. Firing x_{ij} transfers marking M to M' according to

$$M'(p_i, c_{ij}) = M(p_i, c_{ij}) - 1 \quad (2.9)$$

$$M'(q_{ij}) = M(q_{ij}) - 1 \quad (2.10)$$

$$M'(p_j) = M(p_j)$$

$$M'(r_2) = M(r_2)$$

$$M'(r_1) = M(r_1). \quad (2.11)$$

Firing z_{ij} transfers marking M' to M'' according to

$$M''(p_i) = M'(p_i) \quad M''(r_2) = M'(r_2) \quad (2.13)$$

$$M''(r_1) = M'(r_1) + 1 \quad (2.14)$$

$$M''(p_j) = M'(p_j) + 1. \quad (2.15)$$

For transitions t_0 and h_{ij} , the enabling and firing rules for a regular PN can be applied. Thus, this way, the wafer flow with revisiting in cluster tools is exactly modeled by the preceding developed PN. For this PN model, we have the succeeding result [20].

Lemma 2.1: The PN model for dual-arm cluster tools with a wafer flow pattern $(m_1, (m_2, m_3)^k)$ is deadlock free.

According to [5], there are potential deadlocks in dual-arm cluster tools with wafer revisiting if a swap strategy is applied. Notice that a deadlock occurs for this system only in the following case: The robot unloads a raw wafer from the loadlocks, or t_{01} fires, when there is only one robot arm available or $M(r_1) + M(r_2) = 1$. This is totally avoided by the model. Lemma 2.1 indicates that a deadlock is avoided by simply following the operations based on this developed PN model. Notice that the deadlock freeness of the PN is obtained by using macro transitions and defining the transition enabling and firing rules. This clearly demonstrates the advantage of the proposed PN model, as similarly done for robotic systems in [24]. With deadlock freeness, any enabled transition can fire without any problem. Hence, it is helpful in developing a new scheduling method.

As aforementioned, a swap operation at p_i involves the firing of two macro transitions $t_{ki} \in \bullet p_i$ and $t_{ij} \in p_i \bullet$. To initiate a swap operation at p_i , a free arm should move to p_i by firing h_{ij} . Thereafter, for a swap operation, we say that the robot moves to p_i if h_{ij} fires. After the robot waiting in q_{ij} for some time, a wafer in p_i is ready. Thus, x_{ij} fires, which is followed by y_{ki} (the robot rotates) and then z_{ki} . The firing of x_{ij} , y_{ki} , and z_{ki} together completes a swap operation.

E. Modeling Activity Times

For the purpose of scheduling a cluster tool, the model should describe the temporal aspect of activities. To do so, both places and transitions in our model are associated with time since they represent some actions that take time. We use α_{ij} and β_{ij} to denote the time taken for unloading a wafer from a PM (loadlock) and loading a wafer into a PM (loadlock), respectively, and μ_{ij} to denote the time taken for the robot moving from a PM to another with a wafer carried. As pointed out in [4], because the robot move times from one PM to another are very short, they are assumed to be identical. Thus, as done in [4], [5], [35], and [36], it is assumed that $\alpha_{ij} = \alpha$, $\beta_{ij} = \beta$, and $\mu_{ij} = \mu$; and the time needed for the robot to move from a PM to another without carrying a wafer is the same as that with a wafer carried, or the time needed for firing h_{ij} is also μ . When h_{ii} represents the virtual activity of the robot arm moving from PM_i to PM_i , the time is zero. In a wafer revisiting process, each time a wafer visits the same step, the

TABLE I
TIME DURATIONS ASSOCIATED WITH TRANSITIONS AND PLACES

Symbol	Transition or place	Actions	Time duration
α_{ij}	x_{ij}	Robot unloads a wafer from a PM or loadlock	α
μ_{ij}	y_{ij}	Robot moves from a PM to another with or without a wafer carried	μ
β_{ij}	z_{ij}	Robot loads a wafer into a PM or loadlock	β
γ_{ij}	h_{ij}	Robot moves from a PM to another without carrying a wafer	μ
γ_{ii}	h_{ii}	After loading a wafer into a Step i , the robot stays there	0
ω_{ij}	b_{ij}	Robot arm waits in b_{ij} for firing z_{ij} with a wafer held	
φ_{ij}	q_{ij}	Robot arm waits in q_{ij} for firing t_{ij} with no wafer held	
ρ_i	p_i	Wafer is processed in p_i	ρ_i

same operation is performed. Thus, the processing time is the same, too. We use ρ_i to denote the time needed for processing a wafer in p_i . A token $W_d(q)$ in p_i with color c_{ij} can enable t_{ij} only after it stays in p_i for more than ρ_i time units. The times for robot waiting in b_{ij} and q_{ij} are dependent on processing time ρ_i and how the system is scheduled; they are denoted by ω_{ij} and φ_{ij} , respectively. Places r_1 and r_2 and transition t_0 are only used to model the robot availability, and no activity is associated with it. Thus, they take no time. The symbols and time durations for different transitions and places are shown in Table I. The transition firing sequence $x_{ij} \rightarrow y_{ki} \rightarrow z_{ki}$ for a swap operation at p_i can be uninterruptedly and efficiently done, and it may take a time that is different from $\alpha + \mu + \beta$. Let λ_i denote the time needed for a swap operation at p_i and assume, without loss of generality, that $\lambda_i = \lambda$.

Up to now, we complete the PN modeling of a dual-arm cluster tool with wafer revisiting. Based on this PN model, the dynamical behavior of the system is analyzed, and then, a new scheduling strategy is developed to improve its performance next.

III. PROPOSED SCHEDULING STRATEGY

To search for a new scheduling method for performance improvement, it is necessary to make sure how the performance can be improved. Thus, before we present the new scheduling methods, the dynamical behavior of the dual-arm cluster tool with the 3-WC scheduling is examined. For wafer fabrication processes, when a wafer revisits a step, it requires exactly the same processing environment as it visits in its prior step [5]. This is the ALD case. Consequently, each step is composed of only one PM, or $m_1 = m_2 = m_3 = 1$. Note that the results obtained in this paper can be easily extended to the cases with $m_i > 1$, $i \in \mathbf{N}_3 = \{1, 2, 3\}$. The succeeding discussions focus on the situation of $m_1 = m_2 = m_3 = 1$. Thus, the wafer flow pattern is $(1, (1, 1)^2)$ with the revisiting process being $(1, 1)^2$. We assume that PM_i are configured for Step i , $i \in \mathbf{N}_3$.

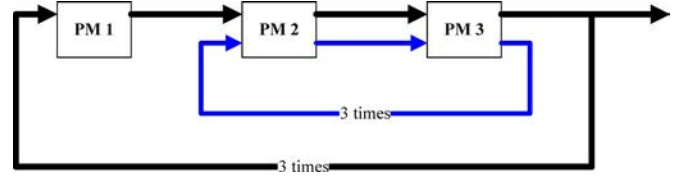


Fig. 5. Wafer processing period with swap strategy.

A. Dynamical Behavior of Dual-Arm Cluster Tools

Let $S_i = \{W_d(k)\}$, $i \in \mathbf{N}_4 = \{1, 2, 3, 4\}$, denote a set of wafers in PM_i or on a robot arm, where d represents the d -th wafer released into the system, and k denotes that the k -th operation is being processed or to be processed. Furthermore, let S_i , $i \in \mathbf{N}_3$, represent the wafers in PM_i , S_4 represent the wafers held by the robot, and the robot is at the PM that processes operation k . Then, a marking of the PN shown in Fig. 4 can be denoted by $M = (S_1, S_2, S_3, S_4)$. Thus, if the 3-WC scheduling is applied to a dual-arm cluster tool with wafer revisiting, the marking evolution of the PN shown in Fig. 4 is $M_1 = (\{W_3(1)\}, \{W_2(2)\}, \{W_1(3)\}, \{W_4(1)\}) \rightarrow M_2 = (\{W_4(1)\}, \{W_3(2)\}, \{W_1(3)\}, \{W_2(3)\}) \rightarrow M_3 = (\{W_4(1)\}, \{W_1(4)\}, \{W_2(3)\}, \{W_3(3)\}) \rightarrow M_4 = (\{W_4(1)\}, \{W_2(4)\}, \{W_3(3)\}, \{W_1(5)\}) \rightarrow M_5 = (\{W_4(1)\}, \{W_3(4)\}, \{W_1(5)\}, \{W_2(5)\}) \rightarrow M_6 = (\{W_4(1)\}, \{W_3(4)\}, \{W_2(5)\}, \{W_5(1)\}) \rightarrow M_7 = (\{W_5(1)\}, \{W_4(2)\}, \{W_3(5)\}, \{W_6(1)\}) \rightarrow M_8 = (\{W_6(1)\}, \{W_5(2)\}, \{W_4(3)\}, \{W_7(1)\}) \rightarrow M_9 = (\{W_7(1)\}, \{W_6(2)\}, \{W_4(3)\}, \{W_5(3)\})$. Notice that, in the sense of PN, markings M_1 and M_8 are equivalent. Thus, it forms a complete period by transforming from M_1 to M_8 .

To transfer M_1 to M_2 (or M_8 to M_9), the following robot activities should be executed: swapping at $p_1 \rightarrow$ moving to Step 2 \rightarrow swapping at $p_2 \rightarrow$ moving to Step 3, which can be described by the transition firing sequence $\sigma_1 = \langle \text{swapping at } p_1 \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow h_{32} \rangle$. To transfer M_2 to M_3 , $\sigma_2 = \langle \text{swapping at } p_3 \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow h_{32} \rangle$ should be executed. Firing sequence σ_2 is repeated for M_3 to M_4 . Similarly, firing sequence $\sigma_3 = \langle \text{swapping at } p_3 \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow h_{30} \rangle$ is for M_4 to M_5 . σ_2 and σ_3 are executed in the revisiting process $(1, 1)^2$ and form a cycle. Because they are done in the revisiting process composed of Steps 2 and 3, we call it a local cycle. Firing sequence $\sigma_4 = \langle \text{swapping at } p_3 \rightarrow y_{30} \rightarrow z_{30} \rightarrow x_{01} \rightarrow y_{01} \rangle$ is for M_5 to M_6 . Firing sequence $\sigma_5 = \langle \text{swapping at } p_1 \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow h_{30} \rightarrow \text{swapping at } p_3 \rightarrow y_{30} \rightarrow z_{30} \rightarrow x_{01} \rightarrow y_{01} \rangle$ is for M_6 to M_7 and M_7 to M_8 , which describes the following robot activities: swapping at $p_1 \rightarrow$ moving to $p_2 \rightarrow$ swapping at $p_2 \rightarrow$ moving to $p_3 \rightarrow$ swapping at $p_3 \rightarrow$ moving to the loadlock \rightarrow dropping a completed wafer into the loadlock \rightarrow unloading a wafer from the loadlock \rightarrow moving to p_1 . Clearly, σ_5 is a robot cycle for the overall system without wafer revisiting. Thus, we call it a global cycle. Notice that σ_1 together with σ_4 is equivalent to σ_5 . From M_8 to M_9 , σ_1 should be executed. Thus, from M_2 to M_9 , there are three local cycles and three global cycles. This implies that, after undergoing three local cycles, the system switches to a global one. Then, after three global cycles, it switches to the local one again, and this process repeats, as illustrated in Fig. 5.

Now, we analyze the dynamical behavior of a dual-arm cluster tool with the 3-WC scheduling. Let Π_i and ψ_1 denote the workload of Step i and the robot cycle time without wafer revisiting and robot waiting, respectively. Then, we have

$$\Pi_i = \rho_i + \lambda, \quad i \in \mathbf{N}_3 \quad (3.1)$$

$$\psi_1 = 3\lambda + 4\mu + \alpha + \beta. \quad (3.2)$$

We use τ_i to denote the sojourn time of a token staying in p_i and $\xi_i = \tau_i - \rho_i$ to denote the delay of the token in p_i . Furthermore, let φ_{ij}^k and ξ_i^k denote the robot waiting time in q_{ij} for the k -th swapping at p_i and the time delay for the k -th wafer completed by p_i , respectively. Let ψ_2 be the robot cycle time for the local cycles. Assume that, at a marking M , it just completes the three local cycles and is going to switch to the global cycles. Then, during the followed three global cycles, the tool can be thought of as a process without wafer revisiting. Without loss of generality, we assume that, during the last local cycle, $\varphi_{23}^k + \varphi_{30}^k > 0$, which must hold when the local cycle operates in its process-bound region. Then, we have the succeeding result [20].

Lemma 3.1: Assume that: 1) the PN model shown in Fig. 4 describes a non-revisiting process, or a token at any marking enables t_{30} only; 2) the current marking is $M = (\{W_3(1)\}, \{W_2(2)\}, \{W_1(3)\}, \{W_4(1)\})$; 3) $\Pi_3 > \Pi_2$ and $\Pi_3 > \psi_1$; 4) starting from M , when the robot moves to p_2 and p_3 , it needs to wait for φ_{23}^1 time units before swapping at p_2 and for φ_{30}^1 before swapping at p_3 , respectively; and 5) starting from $W_4(1)$ entering p_1 , the workload of Step 1 becomes $\Pi_1 > \Pi_3$ or $\chi = \Pi_1 - \Pi_3 > 0$. Then, we have

$$\text{a) } \xi_3^1 = 0 \quad (3.3)$$

$$\text{b) } \xi_3^2 = \max([\chi - (\varphi_{23}^1 + \varphi_{30}^1)], 0) \quad (3.4)$$

$$\text{c) } \xi_3^3 = \begin{cases} 0, & \text{if } (\varphi_{23}^1 + \varphi_{30}^1) > 2\chi \\ 2\chi - (\varphi_{23}^1 + \varphi_{30}^1), & \text{if } \chi < (\varphi_{23}^1 + \varphi_{30}^1) \leq 2\chi \\ \chi, & \text{if } (\varphi_{23}^1 + \varphi_{30}^1) \leq \chi. \end{cases} \quad (3.5)$$

It follows from Lemma 3.1 that, during the three global cycles, the wafer delay ξ_3^k after the wafer is processed in PM_3 is increased cycle by cycle. This can be explained as follows. For the local cycles, the workload Π_3 for PM_3 dominates the process such that $\varphi_{30}^k > 0$ always holds. This implies that, whenever a free robot arm arrives at PM_3 for unloading a wafer being processed, the wafer is not completed yet. Thus, there is no delay in unloading the wafer, or $\xi_3^k = 0$ always holds. However, when the system switches from the local cycles to the global cycles, the workload Π_1 for PM_1 dominates the process such that φ_{12}^k increases cycle by cycle, which may result in $\varphi_{30}^k = 0$. By $\varphi_{30}^k = 0$, it means that, when a free robot arm arrives at PM_3 for unloading a wafer, the wafer is already completed, or $\xi_3^k \geq 0$, or there may be a delay. This way, the system may keep in a transient state but never reaches its steady state. Notice that, with wafer revisiting, the workloads for PM_1 , PM_2 , and PM_3 are Π_1 , $2\Pi_2$, and $2\Pi_3$, respectively. For cluster tools, the robot moving times from one PM to another are much shorter than the wafer processing times [4] such that a

cluster tool often operates in the process-bound region. Thus, if $\Pi_3 \geq \Pi_2$ and $2\Pi_3 \geq \Pi_1$, the lower bound of cycle time for a dual-arm cluster tool with revisiting is $2\Pi_3$. It follows from the preceding analysis that, if $\xi_3^k \geq 0$, the lower bound cannot be reached by the 3-WC scheduling. Letting $\Pi_{3\text{-WC}}$ be the cycle time obtained by the 3-WC scheduling and $\Pi_{\text{local}} = \max(\Pi_3, \Pi_2, \psi_2)$, we have the succeeding results [20].

Lemma 3.2: For a dual-arm cluster tool with wafer revisiting, by using the 3-WC scheduling, if $\Pi_1 \leq 3\Pi_{\text{local}} + \psi_1$ and $\Pi_3 > \Pi_2$, the cycle time is

$$\Pi_{3\text{-WC}} = 2\Pi_{\text{local}} + (\xi_3^1 + \xi_3^2 + \xi_3^3) / 3. \quad (3.6)$$

Lemma 3.3: For a dual-arm cluster tool with wafer revisiting, by using the 3-WC scheduling, if $\Pi_1 \leq 3\Pi_{\text{local}} + \psi_1$, $\Pi_3 > \Pi_2$, $\Pi_1 - \Pi_3 = \chi$, and $\Pi_3 > \psi_1$, the cycle time is

$$\Pi_{3\text{-WC}} = \begin{cases} 2\Pi_3, & \text{if } \Pi_3 - \psi_1 \geq 2\chi \\ 2\Pi_3 + (2\chi + \psi_1 - \Pi_3) / 3, & \text{if } \Pi_3 - \psi_1 < 2\chi. \end{cases} \quad (3.7)$$

It follows from Lemmas 3.2 and 3.3 that, when $\Pi_3 - \psi_1 < 2\chi$, $\xi_3^1 + \xi_3^2 + \xi_3^3 = 2\chi + \psi_1 - \Pi_3$, and the lower bound of cycle time cannot be reached by the 3-WC scheduling. Thus, one needs to search for a new scheduling method to improve the performance under such a condition. It follows from the preceding analysis that it is the time delay $\xi_3^1 + \xi_3^2 + \xi_3^3$ that occurs at Step 3 (when $\Pi_3 > \Pi_2$), during the three global cycles, that deteriorates the performance obtained by the 3-WC scheduling. Furthermore, we have $\xi_3^3 \geq \xi_3^2 \geq \xi_3^1$, or ξ_3^3 makes the most contribution to the delay. Thus, if we can schedule a dual-arm cluster tool with wafer revisiting such that there are less local and global cycles, the performance can be improved. The preceding analysis is conducted under the assumption that $\Pi_3 > \Pi_2$. The result may be different when $\Pi_3 < \Pi_2$. Hence, two scheduling methods for different cases are proposed. They result in a schedule that contains two local cycles and two global cycles, and two wafers are completed in a period. Hence, they are called 2-WC scheduling methods. To obtain a 2-WC schedule, the key is to schedule the local cycles such that the local cycle time is minimized. Based on this observation, the two methods are presented in the succeeding discussion.

B. Schedule 1

Assume that $\Pi_3 > \Pi_2$ and $\rho_3 \geq \rho_2 + 2\mu + \alpha + \beta$. Based on the PN shown in Fig. 4, the system is scheduled such that the marking evolves as follows: $M_1 = (\{W_4(1)\}, \{W_3(2)\}, \{W_2(5)\}, \{W_5(1)\}) \rightarrow M_2 = (\{W_5(1)\}, \{W_4(2)\}, \{W_3(3)\}, \{\}) \rightarrow M_3 = (\{W_5(1)\}, \{W_4(4)\}, \{W_3(5)\}, \{\}) \rightarrow M_4 = (\{W_5(1)\}, \{W_4(4)\}, \{W_3(5)\}, \{W_6(1)\}) \rightarrow M_5 = (\{W_6(1)\}, \{W_5(2)\}, \{W_4(5)\}, \{W_7(1)\}) \rightarrow M_6 = (\{W_7(1)\}, \{W_6(2)\}, \{W_5(3)\}, \{\})$. Markings M_1 and M_5 and M_2 and M_6 are, in fact, equivalent. Thus, a period forms from M_1 to M_5 or from M_2 to M_6 . This evolution can be shown by the changes of token distribution in the PN model. However, to do so, for each marking, it needs a PN. Thus, to save space, we use a graph to illustrate the marking evolution for this schedule, as shown in Fig. 6.

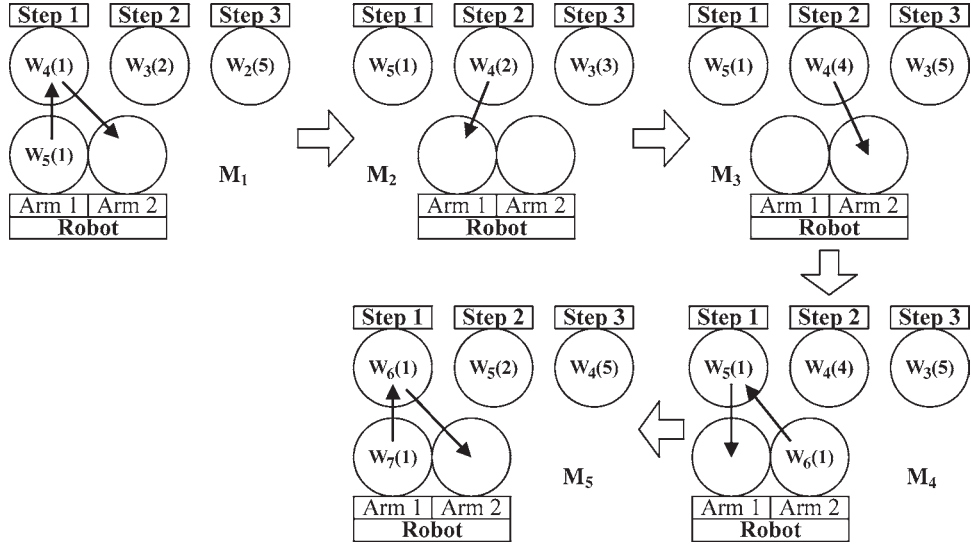


Fig. 6. Illustration of the marking evolution for Schedule 1.

From M_1 to M_2 , the transition firing sequence is $\sigma_1 = \langle \text{swapping at } p_1 \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow h_{30} \rightarrow \text{swapping at } p_3 \rightarrow y_{30} \rightarrow z_{30} \rangle$ that is implemented by the following robot activities: swapping at $p_1 \rightarrow$ moving to $p_2 \rightarrow$ swapping at $p_2 \rightarrow$ moving to $p_3 \rightarrow$ swapping at $p_3 \rightarrow$ moving to the loadlock \rightarrow dropping a completed wafer into the loadlock. These activities take $3\lambda + 3\mu + \beta$ time units. From M_2 to M_3 , $\sigma_2 = \langle h_{23} \rightarrow x_{23} \rightarrow h_{32} \rightarrow \text{swapping at } p_3 \rightarrow y_{32} \rightarrow z_{32} \rightarrow h_{22} \rightarrow \text{waiting in } q_{23} \rightarrow x_{23} \rightarrow h_{32} \rightarrow \text{swapping at } p_3 \rightarrow y_{32} \rightarrow z_{32} \rangle$ that completes two local cycles. From M_3 to M_4 , $\sigma_3 = \langle h_{01} \rightarrow x_{01} \rightarrow h_{12} \rangle$, and it takes $2\mu + \alpha$ time units. From M_4 to M_5 , $\sigma_4 = \langle \text{swapping at } p_1 \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow h_{30} \rightarrow \text{swapping at } p_3 \rightarrow y_{30} \rightarrow z_{30} \rightarrow x_{01} \rightarrow h_{12} \rangle$ that forms a global cycle. From M_5 to M_6 , $\sigma_5 = \langle \text{swapping at } p_1 \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow h_{30} \rightarrow \text{swapping at } p_3 \rightarrow y_{30} \rightarrow z_{30} \rangle$ that is same as σ_1 . Notice that σ_3 and σ_5 together form a global cycle, too. Thus, σ_3 , σ_4 , and σ_5 together form two global cycles. Therefore, $\sigma_2 - \sigma_5$ contain two local cycles and two global cycles with two wafers completed, which is called as a 2-WC schedule. By assumption, $\Pi_3 > \Pi_2$ and $\rho_3 \geq \rho_2 + 2\mu + \alpha + \beta$; then, in σ_2 , each time h_{32} fires, the token in p_3 is ready to fire t_{32} . It implies that no token delay occurs in p_3 . Hence, σ_2 takes $2\Pi_3$ time units. In σ_2 , by a transition firing sequence $\langle \text{swapping at } p_3 \rightarrow y_{32} \rightarrow z_{32} \rangle$, it implies that the time when token $W_3(5)$ enters p_3 is $\mu + \beta$ time units earlier than that when token $W_4(4)$ enters p_2 . Observing Fig. 6, we can find that the key for this schedule is that a swap strategy is not used at Step 2, but it is applied at Step 3.

C. Schedule 2

Assume that $\Pi_2 > \Pi_3$ and $\rho_2 \geq \rho_3 + 2\mu + \alpha + \beta$. Based on the PN shown in Fig. 4, the system is scheduled such that the marking evolves as follows: $M_7 = (\{W_4(1)\}, \{W_3(2)\}, \{W_2(5)\}, \{W_5(1)\}) \rightarrow M_8 = (\{W_5(1)\}, \{W_4(2)\}, \{W_3(3)\}, \{\}) \rightarrow M_9 = (\{W_5(1)\}, \{W_4(4)\}, \{W_3(5)\}, \{\}) \rightarrow M_{10} = (\{W_5(1)\}, \{W_4(4)\}, \{W_3(5)\}, \{W_6(1)\}) \rightarrow M_{11} = (\{W_6(1)\}, \{W_5(2)\}, \{W_4(5)\}, \{W_7(1)\}) \rightarrow M_{12} = (\{W_7(1)\}, \{W_6(2)\},$

$\{W_5(3)\}, \{\})$. Markings M_7 and M_{11} and M_8 and M_{12} are, in fact, equivalent. Thus, a period forms from M_7 to M_{11} or from M_8 to M_{12} . The marking evolution for this schedule is illustrated in Fig. 7.

From M_7 to M_8 , the transition firing sequence is $\sigma_7 = \sigma_1$. From M_8 to M_9 , $\sigma_8 = \langle h_{32} \rightarrow x_{32} \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow y_{23} \rightarrow z_{23} \rightarrow h_{33} \rightarrow \text{waiting in } q_{32} \rightarrow x_{32} \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rightarrow y_{23} \rightarrow z_{23} \rangle$ that completes two local cycles. From M_9 to M_{10} , the transition firing sequence is $\sigma_9 = \sigma_3$. From M_{10} to M_{11} , $\sigma_{10} = \sigma_4$ that forms a global cycle. From M_{11} to M_{12} , the transition firing sequence is $\sigma_{11} = \sigma_1$. It can be seen that the difference between Schedules 1 and 2 is σ_2 and σ_8 . However, the performance is different. Similar to Schedule 1, σ_9 and σ_{11} together form a global cycle, too. Thus, σ_9 , σ_{10} , and σ_{11} together form two global cycles. Therefore, $\sigma_8 - \sigma_{11}$ contain two local cycles and two global cycles with two wafers completed, leading to a 2-WC schedule. In σ_7 and σ_8 , by transition firing sequence $\langle \text{swapping at } p_2 \rightarrow h_{30} \rightarrow \text{swapping at } p_3 \rightarrow y_{30} \rightarrow z_{30} \rightarrow h_{32} \rangle$, it implies that the time when token $W_4(2)$ enters p_2 is $\mu + \lambda$ time units earlier than that when token $W_3(3)$ enters p_3 , but $W_3(3)$ is unloaded from p_3 before $W_4(2)$ from p_2 . This may cause a token delay in p_2 during the local cycles. For this schedule, a swap strategy is not used at Step 3, but it is applied at Step 2.

Notice that the difference between the methods presented in this paper and the 3-WC scheduling one is in scheduling the revisiting process. For Schedule 1, a swap strategy is not applied at Step 2 due to $\Pi_3 > \Pi_2$ and nor at Step 3 when $\Pi_2 > \Pi_3$. Hence, when a revisiting process contains more steps, we can deal with it similarly. If there is more than one revisiting process, we can do the same for each one. If the revisiting times are more than 2, one needs to repeat the revisiting process only. This implies that the presented methods can be extended to other revisiting cases. In addition, both Schedules 1 and 2 are described by operational rules, which are independent of the number of PMs at a step. Thus, it is easy to extend to the cases of $m_i > 1$. Next, the cycle time analysis of the proposed 2-WC schedules is carried out to show the performance improvement.

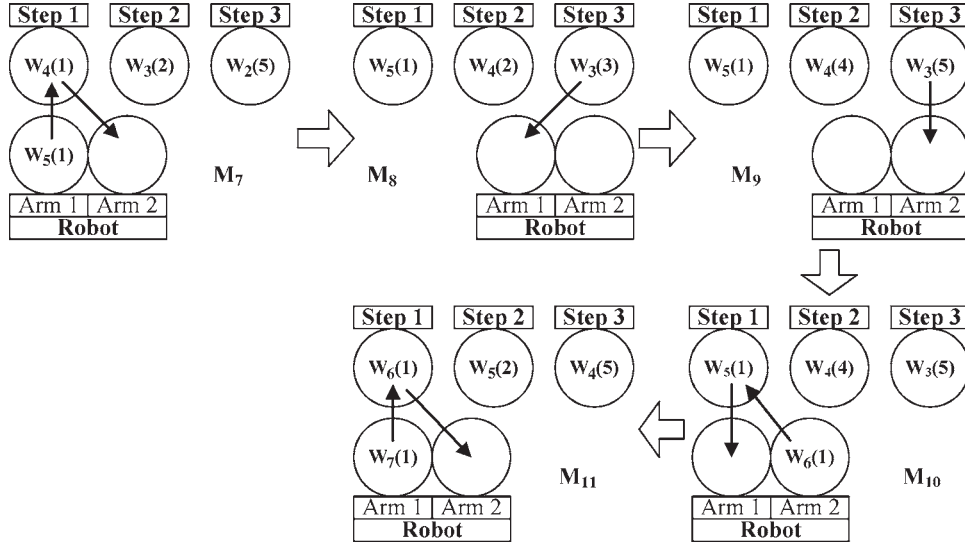


Fig. 7. Illustration of the marking evolution for Schedule 2.

IV. CYCLE TIME ANALYSIS

Because the 2-WC scheduling methods are different from the 3-WC scheduling, the cycle time analysis for the 3-WC scheduling is not applicable. In the prior section, it has been shown that, when $\Pi_{\text{local}} = \max(\Pi_2, \Pi_3) \leq \psi_1$, the 3-WC scheduling can reach the lower bound of cycle time, or it is optimal. Hence, thereafter, in the succeeding discussion, we assume that $\Pi_{\text{local}} > \psi_1$.

When the system reaches marking M_3 in Schedule 1 (M_9 in Schedule 2), it switches from a local cycle to its first global cycle. Then, at M_4 in Schedule 1 (M_{10} in Schedule 2), token $W_6(1)$ is released into the system. For cycle time analysis, it is important to know whether it needs to wait before $W_6(1)$ goes into p_1 .

Theorem 4.1: For a dual-arm cluster tool with wafer revisiting, by using a 2-WC schedule, if $\Pi_1 \leq 2\Pi_{\text{local}} + \psi_1 + \mu$, token $W_6(1)$ at M_4 in Schedule 1 (M_{10} in Schedule 2) can enter p_1 without any waiting.

Proof: Observing the marking evolution of the PN model obtained by Schedules 1 and 2, we can see that, during the transition firing process of $\sigma_1 \rightarrow \sigma_2 \rightarrow \sigma_3$ for Schedule 1 ($\sigma_7 \rightarrow \sigma_8 \rightarrow \sigma_9$ for Schedule 2), token $W_5(1)$ stays in p_1 . Transition firing sequences $\sigma_1 = \sigma_7$ and $\sigma_3 = \sigma_9$ take $3\lambda + 3\mu + \beta$ and $2\mu + \alpha$ time units, respectively, and both σ_2 and σ_8 complete two local cycles. Thus, if $\Pi_1 \leq 2\Pi_{\text{local}} + (3\lambda + 3\mu + \beta) + (2\mu + \alpha) = 2\Pi_{\text{local}} + \psi_1 + \mu$, when token $W_6(1)$ at M_4 in Schedule 1 (M_{10} in Schedule 2) reaches p_1 , $W_5(1)$ in p_1 must be ready for unloading. Hence, $W_6(1)$ does not need to wait. ■

Based on Theorem 4.1 and the PN model, we can then analyze the cycle time obtained by Schedule 1 as follows. In σ_4 , let φ_{23}^1 denote the waiting time between “firing h_{23} ” and “swapping at p_2 ” and φ_{30}^1 denote the waiting time between “ h_{30} ” and “swapping at p_3 ,” respectively.

Lemma 4.1: For a dual-arm cluster tool with wafer revisiting, by using Schedule 1, if $\Pi_1 \leq 2\Pi_{\text{local}} + \psi_1 + \mu$ and $\Pi_3 > \Pi_2$, we have

$$\varphi_{23}^1 = \rho_2 - (\lambda + 3\mu + \alpha) \quad (4.1)$$

$$\varphi_{30}^1 = (\rho_3 - \rho_2) - (\lambda + 2\mu + \beta). \quad (4.2)$$

Proof: Consider the marking evolution of the PN model obtained by Schedule 1. At M_3 , $W_4(4)$ just enters p_2 . By σ_3 and σ_4 , after firing sequence $\sigma_A = \langle h_{01} \rightarrow x_{01} \rightarrow h_{12} \rightarrow \text{swapping at } p_1 \rightarrow h_{23} \rangle$, the robot comes to p_2 for unloading $W_4(4)$. From $\Pi_1 \leq 2\Pi_{\text{local}} + \psi_1 + \mu$ and Theorem 4.1, there is no robot waiting before “swapping at p_1 .” Thus, σ_A takes $\lambda + 3\mu + \alpha$ time units, and it needs to wait for $\rho_2 - (\lambda + 3\mu + \alpha)$ time units for unloading $W_4(4)$, or (4.1) holds.

Considering σ_4 , we know that when $W_4(4)$ is ready for unloading, then after firing sequence (swapping at $p_2 \rightarrow h_{30}$), the robot comes to p_3 for unloading $W_3(5)$. (swapping at $p_2 \rightarrow h_{30}$) takes $\lambda + \mu$ time units. Now, consider σ_2 . By transition firing sequence (swapping at $p_3 \rightarrow y_{32} \rightarrow z_{32}$), it implies that the time when token $W_3(5)$ enters p_3 is $\mu + \beta$ time units earlier than that when token $W_4(4)$ enters p_2 . Thus, in σ_4 , before “swapping at p_2 ” can be executed, it needs to wait for $(\rho_3 - \rho_2) - (\lambda + 2\mu + \beta)$ time units, or (4.2) holds. ■

As for the 3-WC scheduling, the robot waiting time is crucial to analyze the cycle time. Lemma 4.1 provides a method to calculate it for Schedule 1. It should be pointed out that φ_{30}^1 may be negative. In this case, it delays a token in p_3 in the local cycles and makes cycle time $\Pi_{2\text{-WC}}$ longer. With Theorem 4.1 and Lemma 4.1, we can obtain the cycle time for Schedule 1. Without loss of generality, we assume that $\rho_2 - (\lambda + 3\mu + \alpha) > 0$ holds and let $\Pi_{2\text{-WC}}$ denote the cycle time obtained by a 2-WC schedule. Furthermore, let $\xi_1 = \max\{\chi - (\rho_2 - (\lambda + 3\mu + \alpha)), 0\} + (\lambda + 2\mu + \beta) - (\rho_3 - \rho_2)$ and $\xi_2 = \max\{\chi + 2\lambda + 5\mu + \alpha + \beta - \rho_3, 0\}$.

Theorem 4.2: For a dual-arm cluster tool with wafer revisiting, by using Schedule 1, assume that: 1) $\Pi_1 \leq 2\Pi_{\text{local}} + \psi_1 + \mu$; 2) $\Pi_3 > \Pi_2$; and 3) $\chi = \Pi_1 - \Pi_3$. Then, the minimum cycle time is

$$\Pi_{2\text{-WC}} = \begin{cases} 2\Pi_3 + [\max\{-\varphi_{30}^1, 0\}]/2, & \text{if } \chi \leq 0 \\ 2\Pi_3 + \xi_1/2, & \text{if } \chi > 0 \text{ and } \varphi_{30}^1 < 0 \\ 2\Pi_3 + \xi_2/2, & \text{if } \chi > 0 \text{ and } \varphi_{30}^1 > 0. \end{cases} \quad (4.3)$$

Proof: By $\Pi_1 \leq 2\Pi_{\text{local}} + \psi_1 + \mu$ and Theorem 4.1, token $W_6(1)$ at M_4 can enter p_1 without any waiting. Thus, both (4.1) and (4.2) hold, and the result of Lemma 3.1 can be applied. If $\chi \leq 0$, we have $\xi_3^1 = \max\{-\varphi_{30}^1, 0\}$ and $\xi_3^2 = 0$, and $\Pi_{2\text{-WC}} = 2\Pi_3 + (\xi_3^1 + \xi_3^2)/2 = 2\Pi_3 + \max\{-\varphi_{30}^1, 0\}/2$. If $\chi > 0$ and $\varphi_{30}^1 < 0$, we have $\xi_3^1 = -\varphi_{30}^1$ and $\xi_3^2 = \max\{[\chi - \varphi_{23}^1], 0\} = \max\{[\chi - (\rho_2 - (\lambda + 3\mu + \alpha))], 0\}$. Hence, $\Pi_{2\text{-WC}} = 2\Pi_3 + \xi_1/2$. If $\chi > 0$ and $\varphi_{30}^1 \geq 0$, we have $\xi_3^1 = 0$ and $\xi_3^2 = \max\{[\chi - \varphi_{23}^1 - \varphi_{30}^1], 0\}$. Thus, $\Pi_{2\text{-WC}} = 2\Pi_3 + (\xi_3^1 + \xi_3^2)/2 = 2\Pi_3 + \xi_2/2$. This shows the theorem. ■

Notice that ξ_3^3 has no effect on $\Pi_{2\text{-WC}}$, and in general, $\xi_3^3 > \xi_3^2$. Thus, often, $\Pi_{2\text{-WC}} < \Pi_{3\text{-WC}}$ holds, when $\Pi_3 - \psi_1 < 2\chi$. It implies that, in this case, the 2-WC scheduling is better than the 3-WC scheduling. Now, we consider the cycle time obtained by Schedule 2.

Lemma 4.2: For a dual-arm cluster tool with wafer revisiting, by using Schedule 2 for $\Pi_2 > \Pi_3$, in firing sequence $\sigma_B = \langle \text{swapping at } p_2 \rightarrow h_{30} \rightarrow \text{swapping at } p_3 \rightarrow y_{30} \rightarrow z_{30} \rightarrow h_{32} \rightarrow x_{32} \rightarrow h_{23} \rightarrow \text{swapping at } p_2 \rangle$ from σ_7 and σ_8 , the waiting time in q_{23} before executing “swapping at p_2 ” is

$$\varphi_{23}^B = \rho_2 - \rho_3 - (2\mu + \lambda + \alpha). \quad (4.4)$$

Proof: By “swapping at p_2 ”, it puts token $W_4(2)$ into p_2 ; then, after $\langle h_{30} \rightarrow \text{swapping at } p_3 \rangle$, token $W_3(3)$ is put into p_3 $\mu + \lambda$ time units later. Then, after the completion of $W_3(3)$ in p_3 and executing $\langle x_{32} \rightarrow h_{23} \rangle$, the robot comes to p_2 for unloading $W_4(2)$. This implies that, from the time when $W_4(2)$ enters p_2 to the time when the robot comes to p_2 , $\rho_2 + (2\mu + \lambda + \alpha)$ time units pass. Thus, it needs $\varphi_{23}^B = \rho_2 - \rho_3 - (2\mu + \lambda + \alpha)$ time units for $W_4(2)$ in p_2 to be ready. ■

It should be pointed out that $\varphi_{23}^B = \rho_2 - \rho_3 - (2\mu + \lambda + \alpha)$ may be negative. In this case, it delays a token in p_3 in the local cycles and makes cycle time $\Pi_{2\text{-WC}}$ longer, just as φ_{30}^1 does.

Lemma 4.3: For a dual-arm cluster tool with wafer revisiting, by using Schedule 2, if $\Pi_1 \leq 2\Pi_{\text{local}} + \psi_1 + \mu$ and $\Pi_3 < \Pi_2$, we have

$$\varphi_{23}^1 = \rho_2 - (\lambda + 4\mu + \alpha + \beta). \quad (4.5)$$

Proof: With the PN model shown in Fig. 4, consider transition firing sequence $\sigma_c = \langle \text{swapping at } p_2 \rightarrow y_{23} \rightarrow z_{23} \rightarrow h_{01} \rightarrow x_{01} \rightarrow h_{12} \rightarrow \text{swapping at } p_1 \rightarrow h_{23} \rangle$ obtained from σ_8, σ_9 , and σ_{10} . By $\langle \text{swapping at } p_2 \rangle$, it puts token $W_4(4)$ into p_2 , and then, by executing transition firing sequence $\langle y_{23} \rightarrow z_{23} \rightarrow h_{01} \rightarrow x_{01} \rightarrow h_{12} \rightarrow \text{swapping at } p_1 \rightarrow h_{23} \rangle$, the robot comes to p_2 again for unloading $W_4(4)$. Because $\Pi_1 \leq 2\Pi_{\text{local}} + \psi_1 + \mu$, it follows from Theorem 4.1 that there is no waiting before $\langle \text{swapping at } p_1 \rangle$. Firing sequence $\langle y_{23} \rightarrow z_{23} \rightarrow h_{01} \rightarrow x_{01} \rightarrow h_{12} \rightarrow \text{swapping at } p_1 \rightarrow h_{23} \rangle$ takes $\lambda + 4\mu + \alpha + \beta$ time units. Thus, it needs to wait for $\varphi_{23}^1 = \rho_2 - (\lambda + 4\mu + \alpha + \beta)$ time units for $W_4(4)$ to be ready. ■

To analyze the cycle time for Schedule 2, Lemmas 4.2 and 4.3 answer how the robot waiting times should be cal-

culated. Because $\Pi_2 > \psi_1$, $\varphi_{23}^1 > 0$ must hold. Due to $\Pi_2 > \Pi_3$, the cycle time obtained by Schedule 2 is independent of φ_{30}^1 . Let $\xi_3 = \max\{[(2\mu + \lambda + \alpha) - (\rho_2 - \rho_3)], 0\}$ and $\xi_4 = \max\{[\chi + (\lambda + 4\mu + \alpha + \beta) - \rho_2], 0\}$. With Lemmas 4.2 and 4.3, we have the succeeding result.

Theorem 4.3: For a dual-arm cluster tool with wafer revisiting, by using Schedule 2, assume that: 1) $\Pi_1 \leq 2\Pi_{\text{local}} + \psi_1 + \mu$; 2) $\Pi_2 > \Pi_3$; and 3) $\chi = \Pi_1 - \Pi_2$. Then, the cycle time is

$$\Pi_{2\text{-WC}} = \begin{cases} 2\Pi_2 + \xi_3/2, & \text{if } \chi \leq 0 \\ 2\Pi_2 + (\xi_3 + \xi_4)/2, & \text{if } \chi > 0. \end{cases} \quad (4.6)$$

Proof: Let ξ_2^1 and ξ_2^2 denote the token delays at p_2 for the first and second global cycles, respectively. Because $\varphi_{23}^1 = \rho_2 - (\lambda + 4\mu + \alpha + \beta) > 0$, it follows from Lemma 3.1 that $\xi_2^1 = 0$ must hold. It is obvious that $\xi_2^2 = 0$ if $\chi \leq 0$. By Lemma 3.1, $\xi_2^2 = \xi_4 = \max\{[\chi + (\lambda + 4\mu + \alpha + \beta) - \rho_2], 0\}$, when $\chi > 0$. If $\varphi_{23}^B = \rho_2 - \rho_3 - (2\mu + \lambda + \alpha) < 0$, it causes a delay at p_2 in a local cycle. This delay is $\xi_3 = \max\{[(2\mu + \lambda + \alpha) - (\rho_2 - \rho_3)], 0\}$. Thus, the cycle time given by (4.6) is obtained. ■

Notice that φ_{23}^1 in Lemma 4.3 is obtained based on Π_2 with $\Pi_2 > \Pi_3$, whereas that in Lemma 4.1 is obtained based on Π_2 with $\Pi_2 < \Pi_3$. Thus, φ_{23}^1 in Lemma 4.3 is often greater than that in Lemma 4.1. This implies that, by Schedule 2, if $\Pi_2 > \Pi_3$, the performance is better than that if $\Pi_2 < \Pi_3$, in general.

Observing the cycle time given in Theorems 4.2 and 4.3, we know that, if $\varphi_{30}^1 = (\rho_3 - \rho_2) - (\lambda + 2\mu + \beta) \geq 0$ and $\varphi_{23}^B = \rho_2 - \rho_3 - (2\mu + \lambda + \alpha) \geq 0$, the cycle time obtained by the 2-WC scheduling is certainly less than or equal to that obtained by the 3-WC scheduling. This implies that 2-WC scheduling methods are superior to 3-WC scheduling methods in this situation.

Let $\Theta = \max\{\Pi_2, \Pi_3\}$ and $\chi = \Pi_1 - \Theta$. By summarizing the preceding discussion, we have the succeeding result.

Theorem 4.4: For a dual-arm cluster tool with wafer revisiting, the scheduling optimality is as presented here.

- 1) If $\Pi_1 \leq 3\Theta + \psi_1$ and $\Theta < \psi_1$, or $\Pi_1 \leq 3\Theta + \psi_1$, $\Theta > \psi_1$, and $\Theta - \psi_1 \geq 2\chi$, a schedule obtained by the 3-WC scheduling is optimal.
- 2) If $\Pi_1 \leq 3\Theta + \psi_1$, $\Theta > \psi_1$, $\Theta - \psi_1 < 2\chi$, and $\rho_3 \geq \rho_2 + 2\mu + \alpha + \beta$ when $\Pi_3 > \Pi_2$, or $\rho_2 \geq \rho_3 + 2\mu + \alpha + \beta$ when $\Pi_2 > \Pi_3$, a schedule obtained by the 2-WC scheduling is optimal.
- 3) If $\Pi_1 \leq 3\Theta + \psi_1$, $\Theta > \psi_1$, $\Theta - \psi_1 < 2\chi$, and $\rho_3 \geq \rho_2 + 2\mu + \alpha + \beta$ when $\Pi_3 > \Pi_2$, or $\rho_2 \geq \rho_3 + 2\mu + \alpha + \beta$ when $\Pi_2 > \Pi_3$, a schedule obtained by the 2-WC scheduling is optimal.
- 4) If $\Pi_1 \leq 3\Theta + \psi_1$, $\Theta > \psi_1$, $\Theta - \psi_1 < 2\chi$, and $\rho_3 < \rho_2 + 2\mu + \alpha + \beta$ when $\Pi_3 > \Pi_2$, or $\rho_2 < \rho_3 + 2\mu + \alpha + \beta$ when $\Pi_2 > \Pi_3$, it is not clear whether the 3-WC or 2-WC scheduling can obtain the optimal schedule.

By the 2-WC scheduling, we require that when $\Pi_3 > \Pi_2$, $\rho_3 \geq \rho_2 + 2\mu + \alpha + \beta$, or when $\Pi_2 > \Pi_3$, $\rho_2 \geq \rho_3 + 2\mu + \alpha + \beta$ to guarantee that no delay is caused in the local cycles.

Because the robot tasks take much less time than that needed for wafer processing, often, these conditions can be satisfied. When these conditions are not satisfied as given in 4) in Theorem 4.4, it is difficult to do cycle time analysis in an analytical way if the 2-WC scheduling is applied. However, simulation can be used to determine which scheduling method is better. To do so, we need to do simulation for a 2-wafer period. This poses no severe computational burden, fortunately.

It should be pointed out that, up to now, in addition to the swap strategy, there is no other scheduling method for dual-arm cluster tools with wafer revisiting. It has been shown that, for some cases, the proposed methods outperform the swap strategy. Thus, this is a significant contribution in this field.

V. ILLUSTRATIVE EXAMPLES

Here, several industrial application examples are presented to show the results obtained in this paper.

Example 1: For a dual-arm cluster tool with wafer revisiting, the wafer flow pattern is $(1, (1, 1)^2)$; the wafer processing times at Steps 1, 2, and 3 are 270, 200, and 250 s, respectively; and $\alpha = \mu = \beta = 3$ s and $\lambda = 8$ s.

We have $\Pi_1 = 278$ s, $\Pi_2 = 208$ s, $\Pi_3 = 258$ s, and $\psi_1 = 42$ s. Because there is wafer revisiting, the workloads for Steps 1, 2, and 3 are 278, 416, and 516 s, respectively, or Step 3 has the maximal workload. By applying the 3-WC scheduling, because $\Pi_3 - \psi_1 = 216 > 2(\Pi_1 - \Pi_3) = 40 = 2\chi$, it follows from (3.7) that $\Pi_{3\text{-WC}} = 516$ s. It reaches the lower bound and, thus, the optimal solution.

By applying a 2-WC scheduling method, we have $\Pi_3 > \Pi_2$, $\rho_3 - \rho_2 = 50 > \lambda + 2\mu + \beta = 17$, and $\xi_2 = \max\{\chi + 2\lambda + 5\mu + \alpha + \beta - \rho_3, 0\} = \max\{-193, 0\} = 0$. Thus, it follows from (4.3) that $\Pi_{2\text{-WC}} = 516$ s. It is also the optimal solution. In this case, both scheduling methods lead to the optimal performance.

Example 2: In Example 1, the wafer processing at Step 1 is changed into 450 s, and the other parameters are unchanged. In addition, Step 3 has the maximal workload. By applying the 3-WC scheduling, we have $\Pi_3 - \psi_1 = 216 < 2(\Pi_1 - \Pi_3) = 400 = 2\chi$, and it follows from (3.7) that $\Pi_{3\text{-WC}} = (516 + 184/3)$ s ≈ 577.33 s. By the 2-WC scheduling, $\Pi_3 > \Pi_2$ and $\rho_3 \geq \rho_2 + 2\mu + \alpha + \beta$, and Schedule 1 is applied. We have $\rho_3 - \rho_2 = 50 > \lambda + 2\mu + \beta = 17$ and $\xi_2 = \max\{\chi + 2\lambda + 5\mu + \alpha + \beta - \rho_3, 0\} = \max\{-13, 0\} = 0$. Thus, it follows from (4.3) that $\Pi_{2\text{-WC}} = 516$ s, the lower bound is reached, and the solution is optimal. In this case, the 2-WC scheduling is better than the swap strategy.

Examples 1 and 2 show how Π_1 affects $\xi_3^1 + \xi_3^2 + \xi_3^3$, the wafer processing delays, if the 3-WC scheduling is adopted.

Example 3: The wafer processing times at Steps 1, 2, and 3 become 80, 35, and 50 s, respectively; and $\alpha = \mu = \beta = 3$ s and $\lambda = 8$ s, as in Example 1.

We have $\Pi_1 = 88$ s, $\Pi_2 = 43$ s, $\Pi_3 = 58$ s, and $\psi_1 = 42$ s. The workloads for Steps 1, 2, and 3 are 88, 86, and 116 s, respectively; or Step 3 has the maximal workload. By applying the 3-WC scheduling, we have $\Pi_3 - \psi_1 = 16 < (\Pi_1 - \Pi_3) =$

$30 = \chi$; it follows from (3.7) that $\Pi_{3\text{-WC}} = (116 + 44/3)$ s ≈ 130.67 s. By the 2-WC scheduling, with $\Pi_3 > \Pi_2$ and $\rho_3 \geq \rho_2 + 2\mu + \alpha + \beta$, Schedule 1 is applied. We have $\rho_3 - \rho_2 = 15 < \lambda + 2\mu + \beta = 17$ and $\xi_1 = \max\{\chi - (\rho_2 - (\lambda + 3\mu + \alpha)), 0\} + (\lambda + 2\mu + \beta) - (\rho_3 - \rho_2) = 17$ s. Thus, it follows from (4.3) that $\Pi_{2\text{-WC}} = (116 + 17/2)$ s = 124.5 s, which is better than that obtained by the swap strategy.

Example 4: The wafer processing times at Steps 1, 2, and 3 become 80, 50, and 35 s, respectively; and $\alpha = \mu = \beta = 3$ s and $\lambda = 8$ s, as in Example 1.

We have $\Pi_1 = 88$ s, $\Pi_2 = 58$ s, $\Pi_3 = 43$ s, and $\psi_1 = 42$ s. The workloads for Steps 1, 2, and 3 are 88, 116, and 86 s, respectively. For this example, Step 2 has the maximal workload. Furthermore, $\chi = \Pi_1 - \Pi_2 = 30$. By the 3-WC scheduling, the cycle time is 130.67 s, which is same as in Example 3. By the 2-WC scheduling, with $\Pi_2 > \Pi_3$ and $\rho_2 \geq \rho_3 + 2\mu + \alpha + \beta$, Schedule 2 is applied. We have $\varphi_{23}^B = \rho_2 - \rho_3 - (2\mu + \lambda + \alpha) = -2 < 0$, $\xi_3 = \max\{[(2\mu + \lambda + \alpha) - (\rho_2 - \rho_3)], 0\} = 2$ s, and $\xi_4 = \max\{[\chi + (\lambda + 4\mu + \alpha + \beta) - \rho_2], 0\} = 6$ s. Thus, it follows from (4.6) that $\Pi_{2\text{-WC}} = 120$ s, which is better than that obtained by the 3-WC scheduling.

Simulation is done to verify the results for the preceding examples. It is shown that the results are correct. Notice that Π_2 in Example 4 is equal to Π_3 in Example 3, whereas Π_2 in Example 3 is equal to Π_3 in Example 4. However, the cycle time obtained is different. Because a wafer visits Step 2 first and then Step 3, the robot waiting time change is different such that the cycle time is different.

VI. CONCLUSION

Revisiting is one of the important features in semiconductor manufacturing industry. For dual-arm cluster tools, if a swap strategy is applied with wafer revisiting, a manufacturing period contains three local cycles and three global cycles with three wafers completed, leading to a 3-WC schedule. For some situations identified in this paper, such a schedule is not optimal. To improve the performance, based on the PN model developed, two scheduling methods are presented for different cases. They enable a manufacturing period to contain two local cycles and two global cycles with two wafers completed, leading to a 2-WC schedule. Under the situations identified in this paper, the performance obtained by a 2-WC schedule is better than that obtained by a 3-WC schedule. Thus, the method presented in this paper and the 3-WC scheduling method can complement each other for scheduling dual-arm cluster tools with wafer revisiting. Notice that, under some conditions, neither of the 2-WC nor 3-WC scheduling methods are optimal. Thus, further research is needed to search for new scheduling methods.

In some wafer fabrication processes, there is wafer residency time constraint requiring that a wafer should be unloaded from its processing module in a designated time after its completion. Such a constraint together with wafer revisiting certainly makes the problem more complicated. It is our future work to conduct research on cluster tool scheduling with both wafer revisiting and wafer residency constraints.

REFERENCES

- [1] M. Bader, R. Hall, and G. Strasser, "Integrated processing equipment," *Solid State Technol.*, vol. 33, no. 5, pp. 149–154, May 1990.
- [2] P. Burggraaf, "Coping with the high cost of wafer fabs," *Semicond. Int.*, vol. 18, no. 38, pp. 45–50, Mar. 1995.
- [3] S. Ding, J. Yi, and M. Zhang, "Multicluster tools scheduling: An integrated event graph and network model approach," *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 3, pp. 339–351, Aug. 2006.
- [4] J.-H. Kim, T.-E. Lee, H.-Y. Lee, and D.-B. Park, "Scheduling analysis of timed-constrained dual-armed cluster tools," *IEEE Trans. Semicond. Manuf.*, vol. 16, no. 3, pp. 521–534, Aug. 2003.
- [5] H.-Y. Lee and T.-E. Lee, "Scheduling single-armed cluster tools with reentrant wafer flows," *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 2, pp. 226–240, May 2006.
- [6] T.-E. Lee, H.-Y. Lee, and Y.-H. Shin, "Workload balancing and scheduling of a single-armed cluster tool," in *Proc. 5th APIEMS Conf.*, Gold Coast, Australia, 2004, pp. 1–15.
- [7] M.-J. Lopez and S.-C. Wood, "Systems of multiple cluster tools: Configuration, reliability, and performance," *IEEE Trans. Semicond. Manuf.*, vol. 16, no. 2, pp. 170–178, May 2003.
- [8] T. K. McNab, "Cluster tools—Part 1: Emerging processes," *Semicond. Int.*, vol. 13, no. 9, pp. 58–63, Aug. 1990.
- [9] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
- [10] B. Newboe, "Cluster tools: A process solution," *Semicond. Int.*, vol. 13, no. 8, pp. 82–88, Jul. 1990.
- [11] 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. Semicond. Manuf.*, vol. 7, no. 3, pp. 369–373, Aug. 1994.
- [12] T. L. Perkinson, R. S. Gyurcsik, and P. K. MacLarty, "Single-wafer cluster tool performance: An analysis of the effects of redundant chambers and revisitation sequences on throughput," *IEEE Trans. Semicond. Manuf.*, vol. 9, no. 3, pp. 384–400, Aug. 1996.
- [13] 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 Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 564–577, Jul. 2012.
- [14] Y.-H. Shin, T.-E. Lee, J.-H. Kim, and H.-Y. Lee, "Modeling and implementing a real-time scheduler for dual-armed cluster tools," *Comput. Ind.*, vol. 45, no. 1, pp. 13–27, May 2001.
- [15] P. Singer, "The driving forces in cluster tool development," *Semicond. Int.*, vol. 18, no. 8, pp. 113–118, Jul. 1995.
- [16] S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, "A steady-state throughput analysis of cluster tools: Dual-blade versus single-blade robots," *IEEE Trans. Semicond. Manuf.*, vol. 10, no. 4, pp. 418–424, Nov. 1997.
- [17] N. Q. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 29, no. 2, pp. 192–204, May 1999.
- [18] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Petri net-based scheduling of single-arm cluster tools with reentrant atomic layer deposition processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 42–55, Jan. 2011.
- [19] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Petri net-based cycle time analysis of dual-arm cluster tools with wafer revisiting and swapping strategy," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 9–13, 2011, pp. 5499–5504.
- [20] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Petri net modeling and cycle-time analysis of dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 43, no. 1, pp. 196–207, Jan. 2013.
- [21] N. Q. Wu and M. C. Zhou, "Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 658–669, Oct. 2001.
- [22] N. Q. Wu and M. C. Zhou, "Modeling and deadlock control of automated guided vehicle systems," *IEEE/ASME Trans. Mechatron.*, vol. 9, no. 1, pp. 50–57, Mar. 2004.
- [23] N. Q. Wu and M. C. Zhou, "Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1193–1202, Dec. 2005.
- [24] N. Q. Wu and M. C. Zhou, "Deadlock resolution in automated manufacturing systems with robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 3, pp. 474–480, Jul. 2007.
- [25] N. Q. Wu and M. C. Zhou, "Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking," *IEEE/ASME Trans. Mechatron.*, vol. 12, no. 1, pp. 63–72, Feb. 2007.
- [26] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Resource-oriented Petri net for deadlock avoidance in flexible assembly systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 56–69, Jan. 2008.
- [27] N. Q. Wu and M. C. Zhou, *System Modeling and Control With Resource-Oriented Petri Nets*. New York: CRC Press, Oct. 2009.
- [28] N. Q. Wu and M. C. Zhou, "A closed-form solution for schedulability and optimal scheduling of dual-arm cluster tools with wafer residency time constraint based on steady schedule analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 2, pp. 303–315, Apr. 2010.
- [29] N. Q. Wu and M. C. Zhou, "Process vs resource-oriented Petri net modeling of automated manufacturing systems," *Asian J. Control*, vol. 12, no. 3, pp. 267–280, May 2010.
- [30] N. Q. Wu and M. C. Zhou, "Colored timed Petri nets for modeling and analysis of cluster tools," *Asian J. Control*, vol. 12, no. 3, pp. 253–266, May 2010.
- [31] N. Q. Wu and M. C. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012.
- [32] N. Q. Wu and M. C. Zhou, "Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 446–454, Apr. 2012.
- [33] J. Yi, S. Ding, D. Song, and M. Zhang, "Steady-state throughput and scheduling analysis of multicluster tools: A decomposition approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 2, pp. 321–336, Apr. 2008.
- [34] M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: Petri Net Approach*. Singapore: World Scientific, 1998.
- [35] W. M. Zuberek, "Timed Petri nets in modeling and analysis of cluster tools," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 562–575, Oct. 2001.
- [36] W. M. Zuberek, "Cluster tools with chamber revisiting—Modeling and analysis using timed Petri nets," *IEEE Trans. Semicond. Manuf.*, vol. 17, no. 3, pp. 333–344, Aug. 2004.
- [37] *Petri Nets in Flexible and Agile Automation*, M. C. Zhou Ed. London: Kluwer, 1995.
- [38] M. C. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. London: Kluwer, 1993.
- [39] M. C. Zhou and M. D. Jeng, "Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 3, pp. 333–357, Aug. 1998.
- [40] R. Zurawski and M. C. Zhou, "Petri nets and industrial applications: A tutorial," *IEEE Trans. Ind. Elec.*, vol. 41, no. 6, pp. 567–583, Dec. 1994.



Naiqi Wu (M'04–SM'05) received the B.S. degree in electrical engineering from Huainan Institute of Technology, Huainan, China, in 1982, and the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1985 and 1988, respectively.

From 1988 to 1995, he was with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. From 1995 to 1998, he was with Shantou University, Shantou, China. In 1998, he joined Guangdong University of Technology, Guangzhou, China, where he is currently a Professor of industrial and systems engineering with the Department of Industrial Engineering, School of Electromechanical Engineering. He has authored or coauthored one book, three book chapters, and more than 40 papers published in international journals. 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 information assurance.

Dr. Wu serves as the Program Committee Chair of the 2013 IEEE International Conference on Networking, Sensing and Control. He served as a Program Committee Member for many international conferences and was the Program Committee Cochair of the 2012 IEEE International Conference on Networking, Sensing and Control. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS; and IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and is the Editor-in-Chief of the *Industrial Engineering Journal*. He was an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C from 2007–2012.



MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in electrical engineering from Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1990.

In 1990, he joined New Jersey Institute of Technology, Newark, where he has been a Professor of electrical and computer engineering. He is also currently a Professor with Tongji University, Shanghai, China. He has authored or coauthored over 480 publications, including 10 books, over 200 journal papers (majority in IEEE TRANSACTIONS), and 18 book chapters. His research interests include Petri nets, sensor networks, web services, semiconductor manufacturing, and transportation and energy systems.

Dr. Zhou is a Life Member of the Chinese Association for Science and Technology, USA, where he served as President in 1999. He is also a Fellow of the American Association for the Advancement of Science. He is the founding Editor of the IEEE Press Book Series on Systems Science and Engineering; an Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING; and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A; IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS; and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.



Feng Chu (M'07–SM'12) received the B.S. degree in electrical engineering from Hefei University of Technology, Hefei, China, in 1986, the M.S. degree from Institut National Polytechnique de Lorraine, Nancy, France, in 1991, the Ph.D. degree in computer science from the University of Metz, Metz, France, in 1995, and the Habilitation for Research Advising degree from the University of Technology of Compiègne, Compiègne, France, in 2006.

She was with Jiangsu University of Technology, Zhengjiang, China, for two years, and with the National Research Institute in Computer Science and Automation, Metz, for four years. From 1999 to 2009, she was an Associate Professor with the University of Technology of Troyes, Troyes, France. In 2009, she joined the Laboratoire Informatique, Biologie Intégrative et Systèmes Complexes (IBISC), Université d'Evry Val d'Essonne, Evry Cédex, France, where she is currently a Full Professor. She has authored or coauthored more than 50 papers in international journals and more than 60 conference papers. She has led or participated in ten research and industrial projects. Her research interests include modeling, analysis, and optimization of complex systems, particularly intelligent transportation systems; and logistic and production systems based on combinatorial optimization, operations research, and Petri nets.

Dr. Chu was the Program Cochair of the IEEE International Conference on Networking, Sensing, and Control (ICNSC) 2012; and the General Cochair of the IEEE ICNSC 2013. She is also a Technical Committee Member of the International Federation of Automatic Control and of the IEEE Systems, Man, and Cybernetics Society and has served on the program committees for international conferences. She is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C.



Chengbin Chu received the B.Sc. degree in electrical engineering from Hefei University of Technology, Hefei, China, in 1985, and the Ph.D. degree in computer science from the University of Metz, Metz, France, in 1990.

From 1987 to 1996, he was with the National Research Institute in Computer Science and Automation, France. From 1996 to 2008, he was a Professor with the University of Technology of Troyes, France, where he was also the Founding Director of the Industrial Systems Optimization Laboratory. From 2006 to 2010, he was an Overseas Visiting Professor and an Overseas Director with the Department of Industrial Engineering, Xi'an Jiaotong University. He is currently a Chair Professor of the Supply Chain Management with Ecole Centrale Paris, Châtenay-Malabry Cedex, France. He is also currently a Visiting Chair Professor with Tongji University, Shanghai, China. He has authored or coauthored 3 books, more than 120 articles in international journals, and many papers in conference proceedings. His research interests are in the areas related to operations research and modeling, analysis, and optimization of supply chain and production systems.

Dr. Chu was nominated "Chang Jiang Scholars Programme" Chair Professor by the Chinese Ministry of Education in 2005. He served as an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION from 2001 to 2004. He is currently an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. He was a recipient of the Robert Faure Award in 1996 (first prize), for his research and application activities. He also received the "1998 Best Transactions Paper Award" from the IEEE Robotics and Automation Society.