Steady-State Throughput and Scheduling Analysis of Multicluster Tools: A Decomposition Approach

Jingang Yi, *Member, IEEE*, Shengwei Ding, Dezhen Song, *Member, IEEE*, and Mike Tao Zhang, *Senior Member, IEEE*

***Abstract*—Cluster tools are widely used as semiconductor manufacturing equipment. While throughput analysis and scheduling of single-cluster tools have been well-studied, research work on multicluster tools is still at an early stage. In this paper, we analyze steady-state throughput and scheduling of multicluster tools. We consider the case where all wafers follow the same visit flow within a multicluster tool. We propose a decomposition method that reduces a multicluster tool problem to multiple independent single-cluster tool problems. We then apply the existing and extended results of throughput and scheduling analysis for each single-cluster tool. Computation of lower-bound cycle time (fundamental period) is presented. Optimality conditions and robot schedules that realize such lower-bound values are then provided using "pull" and "swap" strategies for single-blade and double-blade robots, respectively. For an M-cluster tool, we present $O(M)$ lower-bound cycle time computation and robot scheduling algorithms. The impact of buffer/process modules on throughput and robot schedules is also studied. A chemical vapor deposition tool is used as an example of multicluster tools to illustrate the decomposition method and algorithms. The numerical and experimental results demonstrate that the proposed decomposition approach provides a powerful method to analyze the throughput and robot schedules of multicluster tools.**

***Note to Practitioners*—Modeling and scheduling of cluster tools are critical to improving the productivity and to enhancing the design of wafer processing flows and equipment for semiconductor manufacturing. This paper presents a decomposition method to calculate the maximum throughput and to analyze the robot action schedule for a cluster tool that contains multiple transfer robots. The proposed algorithms utilize and extend the existing results for the single-cluster tool that only has one transfer robot. Buffer modules between two interconnected clusters are treated as either fictitious cassette modules or fictitious process modules. Therefore, we can decompose the interconnected multicluster tool into multiple single-cluster tools. The outcome of this research work provides not only answers to possible maximum throughput for a given cluster tool system but also robot schedules that address how to reach such a maximum throughput. The scheduler can be implemented and run efficiently on the cluster tool computer of a general configuration cluster tool. Comparing with rule-based and simulation-based scheduling methods, the benefits of the proposed analytical approach include better throughput estimation, faster what-if analysis, and optimal scheduling solutions with varying processing times and cluster tool configurations. We have successfully tested the methodology in this paper on dozens of cluster tools at Intel Corporation.**

***Index Terms*—Cluster tool, decomposition, scheduling, semiconductor manufacturing, throughput.**

NOMENCLATURE[1]

$\mathbb{C}_i(\mathbb{C})$	The ith cluster of an M-cluster tool.
$N_i(N)$	Number of process modules (PMs) in $\mathbb{C}_i(\mathbb{C})$.
$m_i(m)$	Number of robot pick/place actions in $\mathbb{C}_i(\mathbb{C})$.
$C_{ij}(C_j)$	Cassette module j, $j = 1, 2$, in $\mathbb{C}_i(\mathbb{C})$.
C^*_{ij}	Fictitious cassette module j, $j = 1, 2$, in $\mathbb{C}_i$.
$P_{ij}(P_j)$	Process module j in $\mathbb{C}_i(\mathbb{C})$.
B_{ij}	The jth, $j = 1, 2$, buffer module between $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$.
B_i	$B_i = \bigcup_j B_{ij}$: Collection of buffer modules between $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$.
BP_{ij}	The jth, $j = 1, 2$, buffer/process module between $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$.
BP_i	$BP_i = \bigcup_j BP_{ij}$: Collection of buffer/process modules (BPM) between $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$.
$\mathbf{S}_i$	Wafer capacity of B_i (BP_i).
$R_i(R)$	Robot in $\mathbb{C}_i(\mathbb{C})$.
$r_i(r)$	Robot R_i (R) type. $r_i(r) = 1$ if R_i is double-blade and $r_i(r) = 2$ single-blade.
$\mathbf{V}$	Wafer visit route in the multicluster tool.
$\mathbf{V}^*_i$	Wafer visit route in a decoupled $\mathbb{C}_i$.
$\mathbf{FP}$	The fundamental period of cluster tools.
$\mathbf{FP}^{d*}_i$	The calculated fictitious fundamental period of decoupled $\mathbb{C}_i$.
$T_i(T)$	The time interval that $R_i(R)$ picks/places a wafer.
$t_{ij}(t_j)$	Processing time at PM P_{ij} (P_j).
t^{BP}_{ij}	Processing time at BPM BP_{ij}, $j = 1, 2$.
$t^R_i(t^R)$	Robot R_i (R) cassette waiting time.
$\pi_i(\pi)$	Robot R_i (R) action schedule in $\mathbb{C}_i(\mathbb{C})$.

Manuscript received July 11, 2006; revised November 19, 2006. This paper was recommended for publication by Associate Editor T. Lee and Editor N. Viswanadham upon evaluation of the reviewers' comments. This paper was presented in part at the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, and in part at the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy.

J. Yi is with the Department of Mechanical Engineering, San Diego State University, San Diego, CA 92182 USA (e-mail: jgyi@mail.sdsu.edu).

S. Ding is with the Department of Industrial Engineering and Operations Research, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: dingsw@cal.berkeley.edu).

D. Song is with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: dzsong@cs.tamu.edu).

M. T. Zhang was with AzFSM (Fab 12/22/32) Industrial Engineering, Intel Corporation, Chandler, AZ 85248 USA. He is now with Submicon Development Center, Spansion Inc., Sunnyvale, CA 94088 USA (e-mail: mike.zhang@spansion.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2007.906678

[1] Notations in parentheses are for the single-cluster tool case.

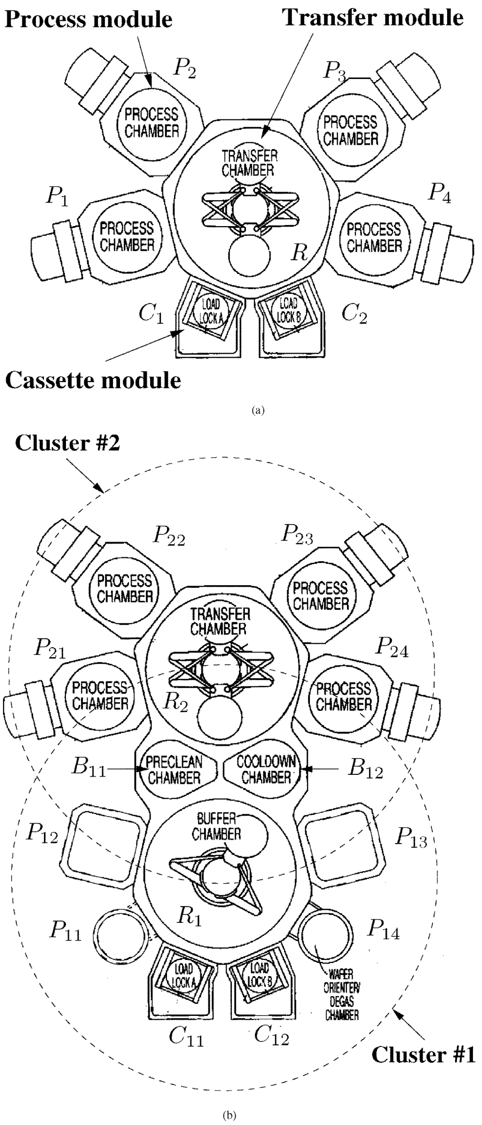


Fig. 1. A schematic of cluster tools [2]. (a) Single-cluster tool. (b) Two-cluster tool.

I. Introduction

CLUSTER tools are widely used as semiconductor manufacturing equipment. In general, a cluster tool is defined as an integrated, environmentally isolated manufacturing system consisting of cassette, process, and transport modules mechanically linked together [1] [Fig. 1(a)]. Cassette modules (CMs) store the unprocessed and processed wafers. Process modules (PMs) execute semiconductor manufacturing processes, such as chemical vapor deposition (CVD), etching, and chemical-mechanical planarization (CMP). Transfer modules (TMs), which are robot manipulators, move wafers among process modules and between process and cassette modules. For a single-cluster tool, only one robot serves multiple process and cassette modules [Fig. 1(a)]. A multicluster tool consists of several single clusters that are interconnected through buffer modules (BMs) [Fig. 1(b)]. During a semiconductor manufacturing process, wafers are transported by robots from the cassette module, sequentially go through various process modules, and then return to the cassette module. Modeling, analysis, and scheduling of cluster tools are critical to improve the productivity.

In this paper, we discuss modeling, analysis, and scheduling of a multicluster tool. We assume that all processing wafers follow the same flow route. Our goal is to find an optimal schedule for TMs that minimizes cycle time and, therefore, maximizes throughput. We consider a general topological connection among the multiple clusters and propose a method to decompose the multicluster tool into multiple individual single-cluster tools. We then extend and apply the existing throughput and scheduling analysis of the single-cluster tool. For an M-cluster tool, $O(M)$ upper-bound maximum throughput computation algorithms are presented. Optimality conditions that could lead to such an upper-bound maximum throughput are then provided and discussed. The impact of combined buffer/process modules (BPMs) on throughput and scheduling of cluster tools is also discussed. A CVD tool is used as an example of the multicluster tools to illustrate the proposed decomposition methods and algorithms.

The contributions of this paper are twofold. First, we formulate a multicluster scheduling and analysis problem, and propose an analytical solution to such a problem using a decomposition method. To our knowledge, there is no research work that formally discusses the maximum throughput calculation and scheduling analysis for a general multicluster tool. The research work presented in this paper not only provides the upper-bound maximum throughput for a given cluster tool system but also discusses optimality conditions and robot schedules to realize such a throughput. Second, using the decomposition approach, the proposed algorithms can help practitioners (such as cluster tool design engineers and process development engineers) to compute and predict the maximum cluster tool throughput. The algorithms can also help identify the process flow bottlenecks, and quickly search for an optimal robot schedule that minimizes the cycle time. The proposed methods and algorithms can be implemented on cluster tool computer (CTC) and be utilized in practice for wafer production. We have successfully applied the proposed methodology in this paper to dozens of cluster tools at Intel Corporation.

The remainder of this paper is organized as follows. We begin with related work in Section II and discuss the structure of the multicluster tools in Section III. In Section IV, we discuss and extend optimal schedules for single-cluster tools. Section V presents a decomposition method for multicluster tools and the algorithms to compute the lower-bound of the minimal cycle time (or so-called fundamental period). Optimality conditions under which the lower-bound fundamental period can be achieved and the robot scheduling algorithm are also presented in this section. In Section VI, we apply and extend decomposition results to analyze BPMs. An example of throughput analysis and robot scheduling is investigated for a CVD tool in Section VII. Finally, we summarize with concluding remarks and future research directions.

II. Related Work

For cluster tools, robot moving and wafer processing sequences repeat cyclically at steady state. Like most literatures, we consider the cycle time for a *one-wafer action sequence* as the optimization objective. A *one-wafer action sequence* is defined as a sequence of robot actions which pick and place each module exactly once [3].

The multicluster tool scheduling problem cannot be simply viewed as a special case of flow-shop scheduling problem or job-shop scheduling problem [4] with deterministic processing and interarrival times. In fact, it is a tight mixed of the two, which makes the problem challenging. The extreme cases of a multicluster tool scheduling problem can be reduced to either case depending on viewpoints. For example, if observing the system from transfer module (robot) viewpoint and assuming the *zero* processing time for each process module, the multicluster tool is now a typical job shop with the robot as the workstation and each pick/place action as the job. Therefore, the scheduling problem is reduced to a typical job shop scheduling problem. If observing the system from wafer viewpoint and assuming the *zero* wafer pick and place action time, the multicluster tool now behaves like a typical flow shop. Each process module can be viewed as individual workstations and there is no buffer in-between workstations. Then, the scheduling reduces to a typical workshop scheduling problem. However, since the processing times are nonzero and the flow between adjacent workstations depends on the availability of robots, the job shop or work shop scheduling results cannot be directly applied to the regular multicluster tool scheduling problem.

In [5] and [6], analytical models of steady-state throughput are discussed for a cluster tool equipped with single-blade and double-blade robots, respectively. A single-blade robot usually can hold only one wafer at a time. A double-blade robot has two independent arms and, therefore, can hold two wafers at the same time with one on each arm.[2] For a cluster tool with a single-blade robot, Perkinson *et al.* [5] propose a "pull" (or so-called downhill) optimal schedule strategy for the robot moving sequence. For a double-blade robot, Venkatesh *et al.* [6] propose the optimal schedule by a "swap" action. Results in [7] and [8] imply that the pull strategy for single-blade cluster tool is *an* optimal schedule. The results presented in [9] for double-grip robotic cells can be applied to a double-blade robot cluster tool and show that the swap schedule is *one* of the optimal strategies. In [10]–[12], scheduling analysis of one robot flow shop is also discussed for the single- and double-gripper robots in a bufferless environment. Recently, Dawande *et al.* [3] summarized the sequencing and scheduling in robotic cells, which is similar to cluster tools.

Petri nets have been used to model the semiconductor manufacturing systems [13]. To model the cluster tool process flows, Srinivasan [14], Zuberek [15], and Wu and Zhou [16] use Petri nets to study the performance of the cluster tool processes for a *given* robot scheduling strategy. For a cluster tool with multiple process modules and transfer robots, Petri nets modeling, scheduling, and analysis can become complicated. Rostami *et al.* [17] and Rostami and Hamidzadeh [18] have used linear programming and heuristic methods to study the optimal schedules for a single-cluster tool with residency constraints on transfer and process modules. Simulation of cluster tools also plays an important role in studying the throughput and in optimizing the process and design, for example, cluster tool physical layout simulation [19]–[21] and event graph modeling and simulation of cluster tools [22]–[24].

Most aforementioned work discuss the steady-state throughput and robot scheduling analysis with an identical wafer flow. Perkinson *et al.* [25] present the impact of parallel (redundant) process modules and revisiting wafer flows on steady-state throughput. Geismar *et al.* [26] extend the result in [25] and discuss the throughput and scheduling analysis of a robotic cell with a single-gripper robot and parallel stations. Herrmann *et al.* [27] study the impact of processing time variations on steady-state throughput and robot scheduling using network flow and simulation models for some simple cluster tools. Ding *et al.* [28] extend the network model in [27] to a multicluster tool.

All of the work above discusses the single-cluster tool configuration except simulation study in [24] and [28]. The single-cluster tool scheduling is relatively straightforward. With the increasing complexity of semiconductor manufacturing processes, multicluster tools are needed to accommodate the industry needs. For a multicluster tool [such as the one shown in Fig. 1(b)], wafer flow modeling and scheduling are apparently more complicated than those of a single-cluster tool because multiple robots can move and transfer wafers simultaneously and coordinately.

Recently, Geismar *et al.* [29] discussed a robotic cell with three single-gripper robots for semiconductor manufacturing with identical robot pick/place time. They present a lower-bound for the cycle time for a robotic cell with parallel machines and multiple robots in a bufferless environment. The authors also compare the throughput of the pull strategy [5] for each robot within the cell with a "longest waiting pair" (LWP) strategy that is employed by the manufacturer. Their simulation study shows that for most cases (87%) the pull strategy achieves the lower-bound performance and improves the throughput significantly. In Ding *et al.* [28], an integrated event graph and network model is used to find *all* optimal schedules for a multicluster tool that can achieve the minimum cycle time. Sensitivity analysis of the processing time variation on the entire tool's throughput is also discussed in [28]. A robust robot sequencing and scheduling could be found if there exist several optimal schedules under the nominal processing time that could lead to the same throughput. In [2] and [30], several rule or priority based heuristic scheduling methods of robot actions of multicluster tools have been discussed. However, there are few analysis and comparison studies of those heuristic methods in terms of optimality.

This paper extends the results in Yi *et al.* [31] and Yi *et al.* [32], and we discuss the optimal schedules *only* based on pull and swap robot strategies. We find the optimality conditions under which the pull and swap robot strategies can achieve the minimum cycle time. Pull and swap strategies are of our interests because they are simple and easily implemented in cluster tools [5]–[8], [10], [11], [29]. The main goal of this study is to analytically investigate the throughput and scheduling of multicluster tools with a *general* configuration. The *general* multicluster tool configuration covers all reported cluster tools in existing work. Even for some cluster tools that do not seemingly look like clustered modules in physical layout, such as the CMP polisher in [22] and [26], we can still apply our results to these tools by abstracting the layout into an M-cluster tool discussed in this paper.

[2] Most robots used in the semiconductor manufacturing industry have either one blade or two blades.

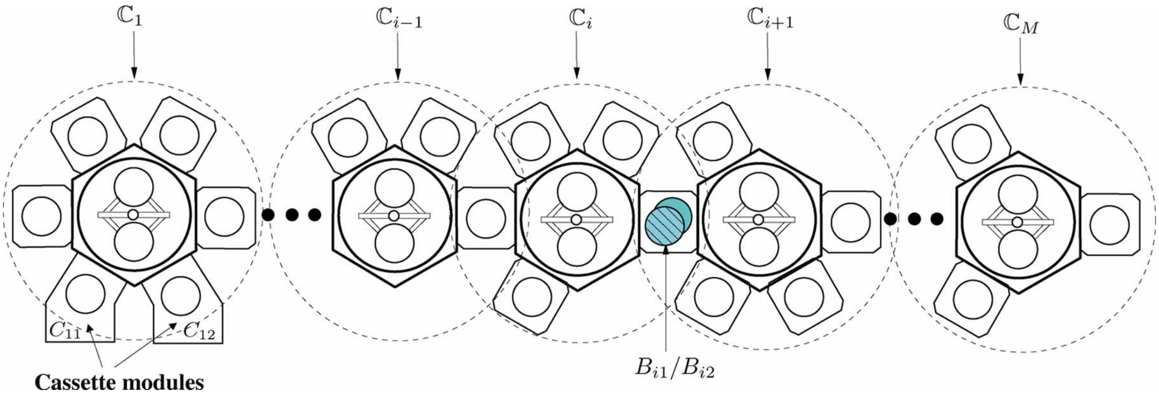


Fig. 2. A schematic of an interconnected M-cluster tool.

III. Problem Description

A. Cluster Tool Assumptions

We consider an interconnected M-cluster tool shown in Fig. 2. For robot action and each PM processing time, we have the following assumptions.

Assumption 1: Processing time assumptions.

1) Robot R_i takes the same amount of time T_i to pick and place a wafer.
2) Robot R_i spends zero time to travel to next module (either PM or CM).
3) Robot transfer time T_i and PM processing time t_{ij} are deterministic.

Assumption 1.2 of zero traveling time among different process modules is reasonable for practical systems since PMs are normally arranged in a circular layout around robot R_i and it, therefore, takes a negligible time for R_i to travel from one PM to another PM. Although in this paper we mainly consider the cases where the robot transferring time is constrained by Assumption 1.1, similar results can be obtained for the cases where robot R_i spends different time intervals to pick and place a wafer.

We also make the following assumptions for the M-cluster tool.

Assumption 2: Multicluster tool assumptions.

1) All wafers follow the identical visit flow $\mathbf{V}$ and this wafer flow $\mathbf{V}$ visits each PM only once.[3]
2) CMs always have wafers/spaces for robot to pick or place at any time.
3) Each robot R_i is either single-bladed or double-bladed.
4) Buffer module B_i has either one- or two-wafer capacity, $\mathbb{C}_i$ and if B_i has two wafer capacity, one is used for wafer outlet direction (relative to $\mathbb{C}_i$) and another is for wafer inlet direction.
5) Each cluster must connect to at least one but at most two other clusters.
6) The multiple clusters cannot form a loop interconnection.

Remark 1: It is noted that in fab production, Assumptions 2.1 is reasonable since a large amount of wafers are processed following the same recipe at one cluster tool. Also, we do not consider the cases where one wafer skips one PM or is processed twice at the same PM (i.e., reentry process flow). Assumption 2.2 is also reasonable since there are multiple front opening unified pods (FOUPs) available at each cluster tools for continuously wafer loading/unloading.

For the M-cluster tool shown in Fig. 2, we have the following definitions.

Definition 1: A BM B_{ij}, $j = 1, 2$, between $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$ is called *buffer/process module* (BPM) if it also functions as a process module with processing time t_{ij}^{BP}.

It is clear that a regular BM can be considered as a BPM with zero processing time. Due to the complexity introduced by BPM, we separate the BPM discussion in Section VI to keep the presentation clarity. We will first focus on multicluster tools interconnected by BMs. Then, we treat BPMs as an extension of the proposed methodology.

Definition 2: A cluster $\mathbb{C}_i$ is called *transfer cluster* if:

1) R_i is a single-blade robot.
2) There is no process module, i.e., $N_i = 0$.
3) Both sides of the BMs have one-wafer capacity, i.e., $\mathbf{S}_{i-1} = 1$ and $\mathbf{S}_i = 1$.

For a transfer cluster, there are not enough wafer storage space to flexibly move wafers within the cluster. We will handle transfer clusters slightly different from regular clusters.

B. Cyclic Wafer Processing

If wafers follow the same visit route within the cluster tool, the production follows the *cyclic production* pattern in which wafers are driven by a fixed sequence of robot actions. For the M-cluster tool, we define that the robot action ACT_i is an atomic *pick* or *place* motion, where $i = 1, \ldots, m$, and m is the total number of robot actions. We can decompose any robot actions into a combination of several such basic atomic motions. For example, after a single-blade robot picks a wafer, a robot place action must be followed. Such a pick/place movement consists of two basic robot actions in sequence. For a double-blade robot, such a constrained movement however is not necessary since a double-blade robot could use one blade to hold a wafer and use the other blade to pick another wafer before placing the wafer on the first blade. Furthermore, for a tighter schedule, it is appropriately assumed that the wafer processing at each PM starts right after the robot places an unprocessed wafer inside that module. Therefore, we can only consider the sequencing and scheduling of robot actions and once such a schedule is fixed the entire cluster tool's activities

[3] We consider multiple parallel process modules as one PM in this assumption.

are determined. We define a *one-wafer cycle* production as follows [3].

Definition 3: A *one-wafer cycle* π is the performance of a feasible one-wafer action sequence that leaves the cluster tool in exactly the same state as its state at the beginning of those actions.

We can denote the robot schedule π as a doublet of its actions ACT_i and their relative starting times ST_i in one cycle: $\pi = \{ACT_i, ST_i\}$, $i = 1, 2, \ldots, m$. We define the function $S(ACT_i, k)$ as the time of completion of the kth cycle ACT_i execution [7]. The *long-run average throughput* or simply, *throughput* can be defined as [7]

$$\mu = \lim_{k \to \infty} \frac{k}{S(ACT_m, k)}. \tag{1}$$

We also take the following definitions from Crama and van de Klundert [7] for *steady state* and *cycle time* $T(\pi)$ for one-wafer cycle π.

Definition 4: A cluster tool repeatedly executing a one-wafer cycle π of robot actions is operating in *steady state* if there exist a constant $T(\pi)$ and a constant $K_0 \in \mathbb{N}$ such that $\forall ACT_i$, $i = 1, \ldots, m$, and $\forall k \in \mathbb{N}$, $k > K_0$, $S(ACT_i, k+1) - S(ACT_i, k) = T(\pi)$. $T(\pi)$ is called the *cycle time* of π.

We define the *optimal schedules* as the set of any one-wafer cycle π_o under which the throughput μ of the cluster tool is maximized. It is observed that if an optimal schedule π_o maximizes the throughput μ, it must minimize the cycle time $T(\pi)$, i.e., $\pi_o = \arg\min_{\pi} T(\pi)$. In this paper, we adopt the terminology "fundamental period" in the literature for the minimal one-wafer cycle time [5], [6].

Definition 5: A fundamental period $\mathbf{FP}$ of a cluster tool is defined as the minimal one-wafer cycle time $T(\pi)$, i.e., $\mathbf{FP} = \min_{\pi} T(\pi)$.

It is obvious that if a cluster tool has a fundamental period $\mathbf{FP}$, then the maximum steady-state throughput is $\mu_{\max} = 1/\mathbf{FP}$.

C. Problem Definition

The objective of this study is to provide a methodology to analyze throughput and robot scheduling of multicluster tools under identical wafer visit routes. For the M-cluster tool under Assumptions 1 and 2, our goal is to address the following questions.

1) What is the steady-state lower-bound fundamental period ($\mathbf{FP}$) (or maximum throughput) of a multicluster tool?
2) What are the optimality conditions and robot schedules for a cluster tool to reach such a minimum cycle time?

IV. Single-Cluster Tool Optimal Schedule

In this section, we first review the existing analysis of scheduling for a single-cluster tool, and then extend these results for the proposed decomposition method in the next section.

A. Maximal Cassette Waiting Time Strategy

Depending on processing time t_j, $j = 1, \ldots, N$, and transfer time T, a single-cluster tool could be running in two possible regions: *process-bound* and *transfer-bound* regions [5], [6]. When the cluster is in a process-bound region, the largest processing time dominates $\mathbf{FP}$ and the robot has some idle time. In the transfer-bound region, processing time is relatively small and the robot is always busy in transferring wafers.

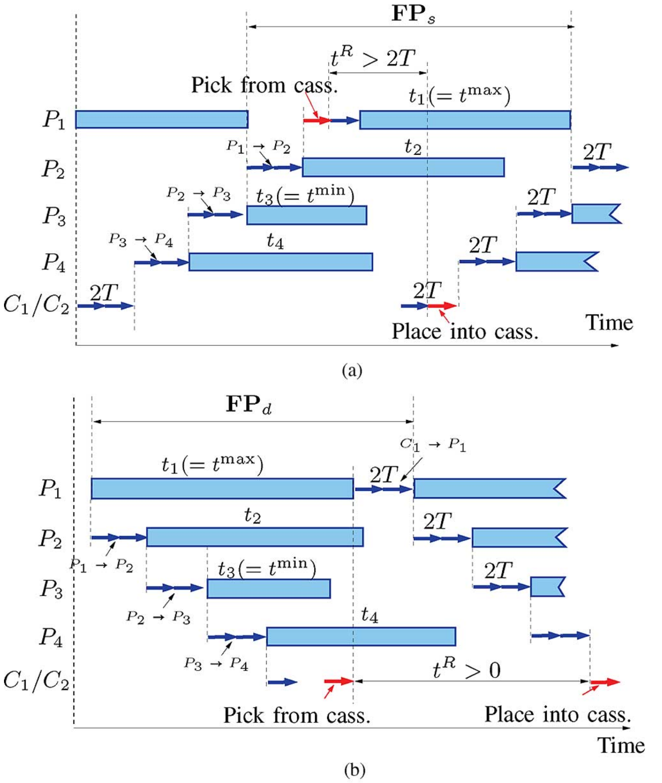


Fig. 3. Gantt chart of maximal cassette waiting time strategies for the example tool shown in Fig. 1(a) that runs in a process-bound (arrows represent robot action durations and rectangle bars represent PM processing durations). (a) Pull strategy for single-blade robot R. (b) Swap strategy for double-blade robot R.

According to [5] and [8], for a single-cluster tool that has N PMs [$N = 4$ for the example shown in Fig. 1(a)], one optimal schedule is given by the following pull robot movements: robot first picks up the wafer in P_N (assuming there is one processed wafer in each PM) and places into C_2, and keeps moving wafers from P_j to P_{j+1}, $j = N-1, \ldots, 1$, until picking up a wafer from cassette C_1 and places it into P_1. Finally, robot waits at P_N to start the next cycle. Fig. 3(a) shows the robot moving sequences and processing sequences for the example cluster tool shown in Fig. 1(a) with a single-blade robot.

For a double-blade robot cluster tool, optimal scheduling is different due to the swap actions that two-blade robot can carry out [6], [10]. Fig. 3(b) shows the robot movment and processing sequences for the cluster tool given in Fig. 1(a) with a double-blade robot. The swap strategy works as follows: the robot first picks an unprocessed wafer from C_1 (on blade 1), moves it to P_1, picks up the processed wafer in P_1 (on blade 2), and places the wafer (on blade 1) into P_1 (swap action). It then moves to P_2 with the wafer from P_1 on blade 2 and start another swap action. The robot keeps swapping wafers through P_j, $j = 1, 2, \ldots, N$, and finally takes the processed wafer from P_N to the cassette.

Denote the maximum and minimum processing times of N process modules as $t^{\max}$ and $t^{\min}$, respectively

$$t^{\max} = \max_{1 \le j \le N} \{t_j\}, \quad t^{\min} = \min_{1 \le j \le N} \{t_j\}. \tag{2}$$

Based on the optimal schedule described above, the fundamental period $\mathbf{FP}_s$ (for a single-blade cluster tool) and $\mathbf{FP}_d$ (for a double-blade cluster tool) can be calculated in a form

$$\mathbf{FP}_s = \begin{cases} 2(N+1)T, & \text{if } t^{\max} < 2(N-1)\,T \\ 4T + t^{\max}, & \text{if } t^{\max} \geq 2(N-1)T \end{cases} \tag{3}$$

$$\mathbf{FP}_d = \begin{cases} 2(N+1)T, & \text{if } t^{\max} < 2N\,T \\ 2T + t^{\max}, & \text{if } t^{\max} \geq 2NT \end{cases}. \tag{4}$$

The first subequation in (3) and (4) represents the case when the tool is running in the transfer-bound region and the second in the process-bound region.

If we consider the cassette module functions as a process module P_0 with zero processing time, i.e., $t_0 = 0$, then we can rewrite (3) and (4) in a compact form that will be used later

$$\mathbf{FP} = 2rT + \max_{1 \leq j \leq N}\{t_j, t_0, 2(N+1-r)T\} \tag{5}$$

where $r = 1$ if R is double-blade and $r = 2$ if R is single-blade.

When the robot schedule discussed above is applied to one of the clusters within a multicluster tool, it is important to analyze the robot waiting time at the interconnected BMs between two adjacent clusters because any waiting time at the BM could result in a delay in one cluster and such a delay could propagate further to other clusters. Equivalently, we need to consider how to allocate the robot idle time (or waiting time) at the cassette modules since for a multicluster tool the BM functions as a cassette module. We define the robot cassette waiting time as follows.

Definition 6: The *robot cassette waiting time* t^R is defined as the time lag of robot R between the moments when finishing the action "pick an unprocessed wafer from input cassette" and starting the subsequent action "place a processed wafer into output cassette."

It is noted that in the single-blade robot schedule discussed above, after the robot moves an unprocessed wafer from cassette C_1 to P_1, robot R may wait for an idle time $t^R - 2T$ [shown in Fig. 3(a)] at cassette C_2 for the next cycle even though the processed wafer in P_4 is ready for pickup. Let t_{idle} denote the total robot idle time. Then

$$t_{\text{idle}} = \max\{t^{\max} - 2(N-1)T, 0\} \tag{6}$$

and

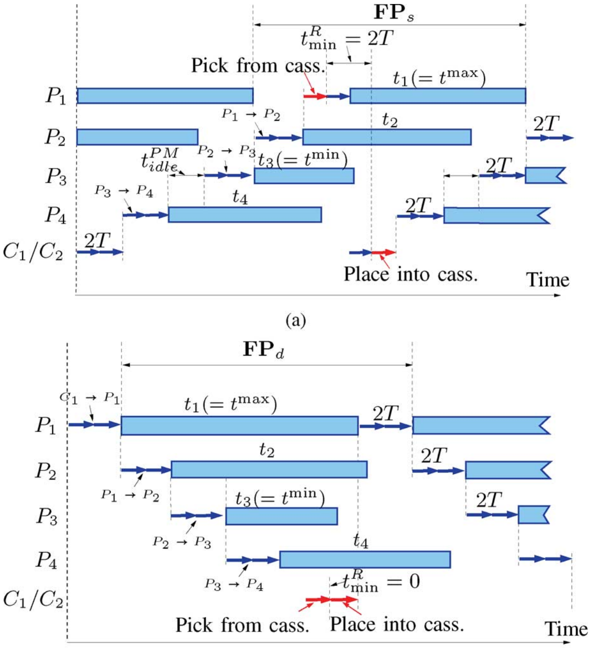
$$t^R = t_{\text{idle}} + 2T. \tag{7}$$

The double-blade robot R also waits at cassette module between pick and place actions at the cassette module [Fig. 3(b)], i.e., $t^R \geq 0$. Therefore, the robot schedule is a maximal cassette waiting time strategy.

In contrast of the maximal cassette waiting time strategy discussed above, we can find alternative pull and swap strategies to minimize the robot cassette waiting time.

B. Minimal Cassette Waiting Time Strategy

For single-blade robots, we consider scheduling robot R to wait as long as possible at the process module with the minimal processing time. We denote the robot waiting time at the PM with the minimum processing time[4] as t_{idle}^{PM}. Let the difference between the maximum and minimum processing times as $\delta t = t^{\max} - t^{\min}$. Then, the maximal robot waiting time at the module with the minimum processing time is



(a)

(b)

Fig. 4. Gantt chart of robot minimal cassette waiting time strategy for the same example in Fig. 1(a) running in a process-bound. (a) Single-blade pull strategy. (b) Double-blade swap strategy.

$$t_{\text{idle}}^{PM} = \min\{\delta t, t_{\text{idle}}\}. \tag{8}$$

For double-blade robots, we can consider not letting R wait any time at the cassette module by forcing the place and the pick actions sequentially [Fig. 4(b)]. This enforcement is feasible since a double-blade robot has two blades and once it places a processed wafer into the cassette the robot could use the other blade to pick an unprocessed wafer.

We propose the following minimal cassette waiting time pull and swap strategies that minimize t^R by allocating the most of t_{idle} to the process module with minimal processing time.

1) Robot R action sequence follows the maximal cassette waiting time pull and swap strategies, respectively.
2) Single-blade robot R waits t_{idle}^{PM} before moving an unprocessed wafer into the PM with processing time $t^{\min}$ [P_3 in Fig. 4(a)].
3) Double-blade robot R places a processed wafer into the cassette right after it picks an unprocessed wafer from the cassette [Fig. 4(b)].

Proposition 1: Using either the minimal cassette waiting time pull strategy (for single-blade robot) or the swap strategy (for double-blade robot), the fundamental period $\mathbf{FP}$ (5) can remain unchanged, while the robot cassette waiting time t^R is minimized as

$$t^R_{\min} = \begin{cases} 0, & \text{if } r = 1 \\ \max\{t^{\min} - 2(N-1)T, 0\} + 2T, & \text{if } r = 2 \end{cases}. \tag{9}$$

[4] If several modules all have the minimum processing time, pick one of these modules.

Proof: See Appendix A. ■

Without confusion, we will abuse notation t^R to denote $t^R_{\min}$ in the rest of this paper unless explicitly indicated.

C. Parallel Process Modules

It is common that there may exist several identically parallel process modules in cluster tools which perform exactly the same functionality. The use of the parallel process modules can prevent or reduce production downtime and increase productivity since wafers only need to go through one of parallel process modules.

We consider a single-cluster tool $\mathbb{C}$ with N process steps. We denote l_i as the redundancy level, i.e., number of parallel modules for process P_i, $i = 1, \ldots, N$. If P_i has only one PM, then $l_i = 1$, namely, no parallel PM. Define the least common multiple (LCM) of l_i as λ, i.e., $\lambda = \mathrm{LCM}(l_1, \ldots, l_N)$. It has been shown in [26] that a pull schedule can still achieve the maximum throughput for a single-blade robot under a set of feasible conditions. We can easily obtain similar results for a double-blade robot case. Therefore, we can extend the fundamental period calculation in (5) for a single-cluster tool with parallel modules as

$$\begin{aligned}\mathbf{FP} &= \max_{1\le i\le N}\left\{\frac{t_i+2rT}{l_i}, t_0+2rT, 2(N+1)T\right\}\\ &= 2rT + \max_{1\le i\le N}\left\{\frac{t_i+2r(1-l_i)T}{l_i}, t_0, 2(N+1-r)T\right\}.\end{aligned} \tag{10}$$

For parallel PM cluster tools, the robot action repeats for each λ-wafer cycle.

V. Optimal Scheduling of Multicluster Tools Using a Decomposition Method

A. Cluster Decomposition Concept

To analyze multicluster systems, we propose an approach to decouple the interconnection among clusters, and then apply the steady-state performance and scheduling results to each decoupled single-cluster tool.

The key of the approach is to decouple the link between clusters. As shown in Fig. 2, for $\mathbb{C}_i$ in a multicluster system, we know that wafers flow in or out of the cluster through either BMs or cassette modules. $\mathbb{C}_i$ exchanges wafers with $\mathbb{C}_{i-1}$ through B_{i-1}, $i > 1$. B_{i-1} plays dual roles: for $\mathbb{C}_i$, B_{i-1} acts like a fictitious cassette module; for $\mathbb{C}_{i-1}$, on the other hand, it acts like a fictitious process module.

Fig. 5 shows an example of how to decouple a two-cluster tool with a two-wafer capacity BM into two single-cluster tools. We consider how to construct the two decoupled single-cluster tools. For decoupled $\mathbb{C}_1$, wafers leave through B_{11} and reenter through B_{12}. So, B_1 are considered as one fictitious process module P^*_{13}.[5] Moreover, the processing time of P^*_{13} depends on the interconnection configurations between $\mathbb{C}_1$ and $\mathbb{C}_2$ and its value will be calculated in later sections. For decoupled $\mathbb{C}_2$, B_{11} and B_{12} become the fictitious cassette modules C^*_{21} and C^*_{22}, respectively. So, we have the second single-cluster tool $\mathbb{C}_2$ with four process modules and two fictitious cassette modules, as shown in Fig. 5.

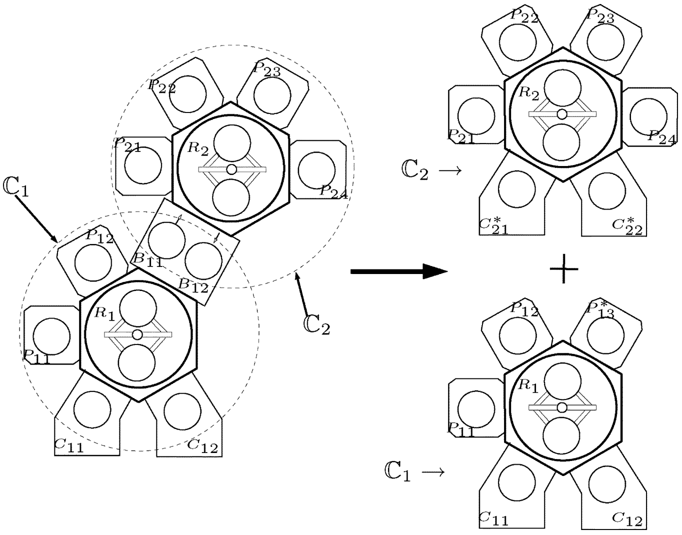


Fig. 5. An example of the decoupling method for an interconnected two-cluster tool shown in Fig. 1(b).

Suppose that the wafer process flow $\mathbf{V}$ for the two-cluster tool shown in Fig. 5 is as follows:

$$\begin{aligned}\mathbf{V}: C_{11} \xrightarrow{R_1} P_{11} \xrightarrow{R_1} P_{12} \xrightarrow{R_1} B_{11} \xrightarrow{R_2} P_{21} \xrightarrow{R_2} P_{22}\\ \xrightarrow{R_2} P_{23} \xrightarrow{R_2} P_{24} \xrightarrow{R_2} B_{12} \xrightarrow{R_1} C_{12}.\end{aligned}$$

Then, after decomposition, the wafer flows for the two single-cluster tools are

$$\mathbf{V}^*_1: C_{11} \xrightarrow{R_1} P_{11} \xrightarrow{R_1} P_{12} \xrightarrow{R_1} P^*_{13} \xrightarrow{R_1} C_{12}$$

and

$$\mathbf{V}^*_2: C^*_{21} \xrightarrow{R_2} P_{21} \xrightarrow{R_2} P_{22} \xrightarrow{R_2} P_{23} \xrightarrow{R_2} P_{24} \xrightarrow{R_2} C^*_{22},$$

respectively.

If two transfer clusters are connected to each other (say $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$ are both transfer clusters), we can combine them as one transfer cluster $\mathbb{C}^*_i$. The transfer cluster $\mathbb{C}^*_i$ consists of a fictitious robot R^*_i and BMs B_{i-1} and B_{i+1} with transfer time $T^*_i = T_i + T_{i+1}$. Thus, it is appropriate to assume that transfer clusters are not connected to each other in series.

B. Computing Lower-Bound Fundamental Period

We consider decoupling a multicluster tool into a set of independently running single-cluster tools by treating BMs as either fictitious cassette modules or fictitious process modules. We then find the minimal fundamental period $\mathbf{FP}^{d*}_i$ for each $\mathbb{C}_i$. After we obtain the set of $\{\mathbf{FP}^{d*}_i\}$, $i = 1, \ldots, M$, we identify the lower-bound value as the largest $\mathbf{FP}^{d*}_i$, which will determine $\mathbf{FP}$ for the entire system.

Assume that a decoupled $\mathbb{C}_i$ has $N_i \ge 0$ PMs and denote the fictitious process module as the $(N_i + 1)$th PM, denoted by $P^*_{i(N_i+1)}$. We assume that $P^*_{i(N_i+1)}$ has a fictitious processing time $t^{d*}_{i(N_i+1)}$ and the fictitious cassette modules C^*_i has a wafer supply time t^{d*}_{i0} (we will discuss it later in details). Assuming

[5] We use the superscript "$*$" to denote the variables associated with fictitious modules.

that t^{d*}_{i0} and $t^{d*}_{i(N_i+1)}$ are known, $i = 1, \ldots, M-1$, from (5), we can obtain $\mathbf{FP}^{d*}_i$ as follows:

$$\mathbf{FP}^{d*}_i = \begin{cases} 4T_i + t^{d*}_{i(N_i+1)} + t^{d*}_{i0}, & \text{if } \mathbb{C}_i \text{ is a transfer cluster} \\ 2r_iT_i + t^{md}_i, & \text{otherwise} \end{cases} \tag{11}$$

where

$$t^{md}_i = \max_{1 \le j \le N_i} \{t_{ij}, t^{d*}_{i(N_i+1)}, t^{d*}_{i0}, 2((N_i+1)+1-r_i)T_i\}.$$

In (11), we have to separate the case when $\mathbb{C}_i$ is a transfer cluster where the fundamental period calculation is different. For $\mathbb{C}_M$, there is no fictitious PM and we can, therefore, ignore t^{d*}_{i0} in the calculation of $\mathbf{FP}^{d*}_M$.

Let us now discuss how to compute t^{d*}_{i0} and $t^{d*}_{i(N_i+1)}$.

Fictitious cassette C^*_i's supply time t^{d*}_{i0} can be considered as the minimum loading delay time of R_{i-1}. Robot R_i cannot pick a wafer from the fictitious cassette module before robot R_{i-1} finishes loading the wafer. This incurs a loading delay t^{d*}_{i0}. For example, if both R_i and R_{i+1} are double-blade robots, swap action is considered as the most efficient moving sequence of a double-blade robot since there is no time gap t^R between picking and placing actions, thus $t^{d*}_{i0} = 0$ in this case. We consider the loading delay t^{d*}_{i0} as how long it takes R_i to refill C^*_{i+1} after R_{i+1} places a wafer into C^*_{i+1}. C^*_{i+1} has a processing time $T_i - T_{i+1}$ (assuming this value is non-negative) because it will take at least $T_i - T_{i+1}$ between the moment R_{i+1} placing a wafer into C^*_{i+1} and the moment R_i placing a unprocessed wafer for R_{i+1} pickup. Therefore, we can obtain t^{d*}_{i0}, $i > 1$, for a general case as

$$t^{d*}_{i0} = \begin{cases} 2r_{i-1}T_{i-1}, & \mathbf{S}_{i-1} = 1 \\ \max\{T_{i-1} - (2r_i - 1)T_i, 0\}, & \mathbf{S}_{i-1} = 2 \end{cases}. \tag{12}$$

For cluster $\mathbb{C}_1$, we have $t^{d*}_{10} = 0$ due to the fact that $\mathbb{C}_1$ always has real cassette modules with wafers/spaces inside.

The value of $t^{d*}_{i(N_i+1)}$ depends on the minimal loading time delay at $P^*_{i(N_i+1)}$. The minimal loading delay time can be obtained in the same way as we discussed above for t^{d*}_{i0} with additional consideration for minimal robot cassette waiting time t^R_{i+1} of $\mathbb{C}_{i+1}$. Therefore, we can calculate $t^{d*}_{i(N_i+1)}$ as

$$t^{d*}_{i(N_i+1)} = \begin{cases} 2r_{i+1}T_{i+1} + t^R_{i+1}, & \mathbf{S}_i = 1 \\ \max\{T_{i+1} - (2r_i - 1)T_i, 0\}, & \mathbf{S}_i = 2 \end{cases}. \tag{13}$$

With the analysis above, the computation of $\mathbf{FP}$ for the multicluster tools can be described as in Algorithm 1.

Algorithm 1: Decoupled **FP** calculation of a cluster tool.

```
Input : Cluster tool configuration and wafer flow $\mathbf{V}$
Output : Lower-bound fundamental period $\mathbf{FP}$ for $\mathbf{V}$
Decompose the tool into $M$ single-cluster tools;
Construct the wafer flows $\mathbf{V}^*_i, i = 1, 2, \ldots, M$, for each $\mathbb{C}_i$.
for $i = M$ to 1 do
    Construct $t^{d*}_{i0}$ by (12) and $t^{d*}_{i(N_1+1)}$ by (13).
    Calculate $\mathbf{FP}^{d*}_i$ for cluster $\mathbb{C}_i$ using (11).
end
$\mathbf{FP} = \max_{1 \le i \le M} \{\mathbf{FP}^{d*}_i\}$.
```

C. Optimality Conditions of Lower-Bound Fundamental Period

The lower-bound fundamental period $\mathbf{FP}$ computed in the previous section might not be realized for all types of multicluster tool configurations due to the fact that we use a minimal time interaction between two adjacent clusters in the computations. Therefore, it is natural to ask what are the optimality conditions under which the computed $\mathbf{FP}$ is feasible and how to find an optimal robot schedule under these conditions.

Proposition 2: For an M-cluster tool, the computed fundamental period $\mathbf{FP}$ by Algorithm 1 is feasible if for each cluster $\mathbb{C}_k$, $k > 1$, the minimal robot cassette waiting time t^R_k satisfies the following condition:

$$t^R_k \le \begin{cases} \mathbf{FP} - 2r_{k-1}T_{k-1} - 2T_k, & \text{if } \mathbf{S}_{k-1} = 1 \\ 2\mathbf{FP} - 2r_{k-1}T_{k-1} - 2T_k, & \text{if } \mathbf{S}_{k-1} = 2 \end{cases}. \tag{14}$$

Proof: See Appendix B. ■

Remark 2: Optimality conditions in Proposition 2 are sufficient but not necessary. However, for most cluster tools in practice, these optimal conditions are easily satisfied since double-blade robots are widely used in practice (such as the CVD tool that we will discuss in Section VII).

D. Robot Scheduling

In this section, we provide a scheduling algorithm for the multicluster tool that could reach $\mathbf{FP}$ by Algorithm 1. We use a "no waiting" schedule that has been implemented in practice: once the wafer has been placed into the process module, the process starts right away. For such a schedule, each process starting time is completely dependent on the robot action starting time. For robot R_i and decoupled $\mathbb{C}_i$, we denote its schedule as π_i. After a proper timing shift of π_i's by interconnection relationships, they can be fitted into a multicluster schedule $\boldsymbol{\pi}$ with the fundamental period $\mathbf{FP}$. The feasibility of the optimal schedule by this timing shift is guaranteed by the optimality conditions discussed in the previous section.

The decomposed cluster schedule π_i is chosen as follows.

1) If $\mathbb{C}_i$ is a single-blade cluster, time zero in π_i starts at the moment when robot R_i takes a processed wafer from the last process module (followed by the action of placing the same wafer into the fictitious cassette module C^*_i). All other robot actions follow the minimal cassette waiting time pull strategy. The last action is to take an unprocessed wafer from C^*_i. If the last action finishes before $\mathbf{FP}$, the robot keeps idle until $\mathbf{FP}$.
2) If $\mathbb{C}_i$ is a double-blade cluster, the time zero robot action is the moment when R_i picks an unprocessed wafer from C^*_i (followed by the action of placing a processed wafer into C^*_i). All other robot actions follow the swap strategy. The last R_i action is to place a wafer into the last (fictitious) process module P_{iN_i} ($P_{i(N_i+1)^*}$). If the last action finishes before $\mathbf{FP}$, the robot keeps idle until $\mathbf{FP}$.
3) If $\mathbb{C}_i$ is a transfer cluster, the time zero robot action is the moment when R_i picks a processed wafer from C^*_{i+1} (followed by the action of placing the same wafer into C^*_i). Robot R_i then waits for C^*_i to be ready for next unprocessed wafer. Finally, it picks the wafer from C^*_i and places it into C^*_{i+1}. If the last action finishes before $\mathbf{FP}$, the robot keeps idle until $\mathbf{FP}$.

Algorithm 2: A "no-wait" optimal robot scheduling.

```
Input : Cluster tool configuration, wafer flow $\mathbf{V}$, and fundamental period $\mathbf{FP}$
Output: Scheduling $\boldsymbol{\pi}$ for $\mathbf{V}$
Obtain the decomposed schedule $\pi_i = \{ACT_i^j, ST_i^j\}$, $j = 1, \ldots, m_i$, for cluster $\mathbb{C}_i$, $i = 1, \ldots, M$, using (a) swap strategy ($r_i = 1$) or (b) the minimal cassette waiting time pull strategy ($r_i = 2$).
Initialize system schedule as $\boldsymbol{\pi} \leftarrow \pi_1$.
for $i = 2$ to $M$ do
    Search for $ACT_{i-1}^s \in \pi_{i-1}$ that picks wafers from
    $B_{i-1}/BP_{i-1}$. Mark $ACT_{i-1}^s$ starting time as $ST_{i-1}^s$.
    $T_i^{\text{shift}} \leftarrow ST_{i-1}^s - 2T_i$.
    for $j = 1$ to $m_i$ do
        Update $ST_i^j \leftarrow ST_i^j + T_i^{\text{shift}}$.
    end
    $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} + \pi_i$.
end
```

Algorithm 2 describes an optimal robot schedule as discussed above. In Algorithm 2, m_i denotes the number of robot R_i actions of decoupled $\mathbb{C}_i$. This algorithm leads to a unique scheduling solution by forcing all robot movements to be started as late as possible up to when the previous cluster wants to take the wafer from the BM. It is also noted that since the decomposed schedules π_i are extended to $\mathbf{FP}$, all process modules can be fitted into the gaps between robot actions in the schedule.

VI. Buffer/Process Modules (BPM)

The use of BPMs can make the cluster tool more compact and save the tool's footprint and cost. The existence of BPMs could, however, affect the throughput and the robot schedule because its dual role as a process module could introduce a significant complexity in analysis.

A. *Fundamental Period Computation With BPMs*

We consider that there is a BPM BP_k between $\mathbb{C}_k$ and $\mathbb{C}_{k+1}$, $1 \leq k \leq M-2$ (Fig. 6). BP_k has either one- or two-wafer capacity. We denote the incoming BPM (wafer flow from $\mathbb{C}_k$ to $\mathbb{C}_{k+1}$) as BP_{k1} and outgoing BPM (wafer flow from $\mathbb{C}_{k+1}$ to $\mathbb{C}_k$) as BP_{k2}, respectively. If BP_k has one-wafer capacity, $\mathbf{S}_k = 1$, then BP_{k1} and BP_{k2} share the same physical buffer device. If BP_k has two-wafer capacity, $\mathbf{S}_k = 2$, BP_{k1} and BP_{k2} are independent buffer devices. For presentation simplicity, we use Fig. 6 to represent both cases. Let t_{k1}^{BP} and t_{k2}^{BP} be the processing time of BP_{k1} and BP_{k2}, respectively.

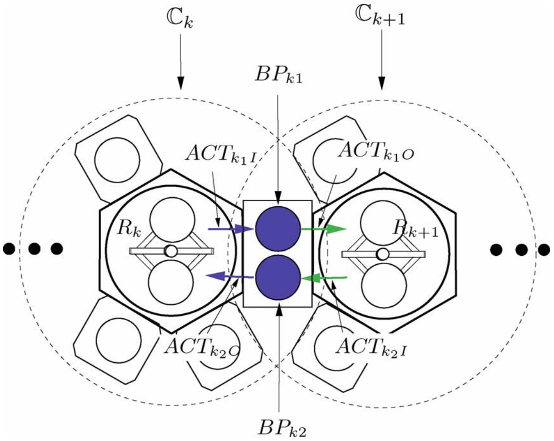


Fig. 6. A combined BPM with two-wafer capacity.

For presentation convenience, we introduce following notations:

$$T_k^s = T_k + T_{k+1}, \quad T_k^B = t_{k+1}^R + t_k^T, \quad t_k^T = 2(r_k - 1)T_k \tag{15}$$

where t_k^R is the minimal robot cassette waiting time for the decoupled cluster $\mathbb{C}_k$.

We first compute the fundamental period by the decomposition algorithm in Section V assuming that there were no BPM within the cluster tool, namely, $t_{k1}^{BP} = t_{k2}^{BP} = 0$. We denote such a calculation as $\mathbf{FP}_0$. Depending on the BPM wafer capacity and processing time t_{k1}^{BP}, t_{k2}^{BP}, we can obtain the following results.

Proposition 3: For an M-cluster tool with a BPM BP_k between clusters $\mathbb{C}_k$ and $\mathbb{C}_{k+1}$, the fundamental period $\mathbf{FP}$ of the cluster tool can be calculated as follows.

- If $\mathbf{S}_k = 1$, see (16) shown at the bottom of the page.
- If $\mathbf{S}_k = 2$

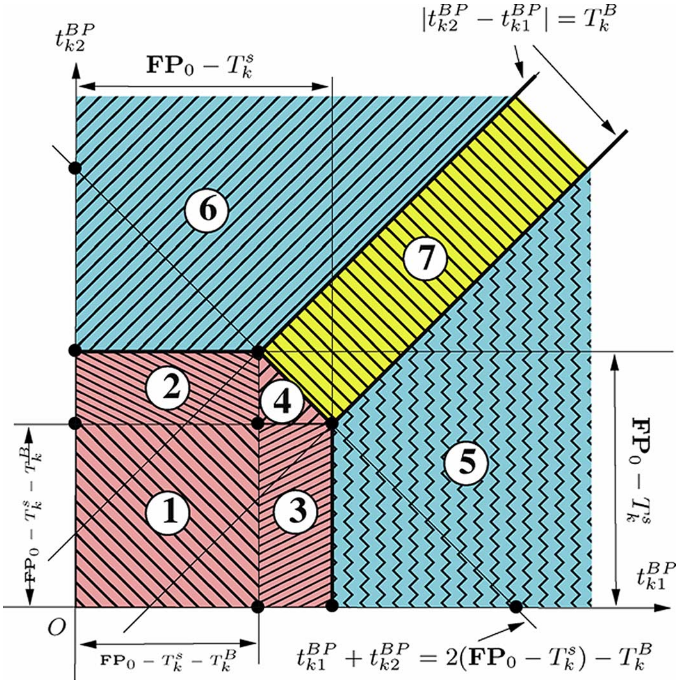
$$\mathbf{FP} = \begin{cases} \mathbf{FP}_0, & \text{if } (t_{k1}^{BP}, t_{k2}^{BP}) \in \bigcup_{i=1}^{4} \mathcal{T}_i \\ \max\{t_{k1}^{BP}, t_{k2}^{BP}\} + T_k^s, & \text{if } (t_{k1}^{BP}, t_{k2}^{BP}) \in \bigcup_{i=5}^{6} \mathcal{T}_i \\ \frac{t_{k1}^{BP} + t_{k2}^{BP} + T_k^B}{2} + T_k^s, & \text{if } (t_{k1}^{BP}, t_{k2}^{BP}) \in \mathcal{T}_7 \end{cases} \tag{17}$$

where $\mathcal{T}_1$-$\mathcal{T}_7$ are defined as (19a)–(19g)[6] (and graphically shown in the t_{k1}^{BP}-t_{k2}^{BP} plane in Fig. 7)

$$\mathcal{T}_1 = \left\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \,\middle|\, t_{ki}^{BP} \leq \mathbf{FP}_0 - T_k^s - T_k^B, i = 1, 2\right\} \tag{19a}$$

$$\mathbf{FP} = \begin{cases} \mathbf{FP}_0, & \text{if } t_{k1}^{BP} + t_{k2}^{BP} \leq \mathbf{FP}_0 - 2T_k^s - T_k^B \\ t_{k1}^{BP} + t_{k2}^{BP} + T_k^B + 2T_k^s, & \text{otherwise} \end{cases} \tag{16}$$

[6] $\mathbb{R}_+$ is defined as the set of non-negative real numbers, i.e., $\mathbb{R} = \{x \in \mathbb{R} |\ x \geq 0\}$.



t_{k2}^{BP}
$\mathbf{FP}_0 - T_k^s$
$|t_{k2}^{BP} - t_{k1}^{BP}| = T_k^B$
6
7
2
4
1
3
5
$\mathbf{FP}_0 - T_k^s - T_k^B$
$\mathbf{FP}_0 - T_k^s$
t_{k1}^{BP}
O
$\mathbf{FP}_0 - T_k^s - T_k^B$
$t_{k1}^{BP} + t_{k2}^{BP} = 2(\mathbf{FP}_0 - T_k^s) - T_k^B$

Fig. 7. **FP** calculation for different BPM process time t_{k1}^{BP} and t_{k2}^{BP} distributions if $\mathbf{S}_k = 2$.

$$\mathcal{T}_2 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k1}^{BP} \le \mathbf{FP}_0 - T_k^s - T_k^B, \text{ and } \mathbf{FP}_0 - T_k^s - T_k^B \le t_{k2}^{BP} \le \mathbf{FP}_0 - T_k^s\Big\} \tag{19b}$$

$$\mathcal{T}_3 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k2}^{BP} \le \mathbf{FP}_0 - T_k^s - T_k^B, \text{ and } \mathbf{FP}_0 - T_k^s - T_k^B \le t_{k1}^{BP} \le \mathbf{FP}_0 - T_k^s\Big\} \tag{19c}$$

$$\mathcal{T}_4 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| \mathbf{FP}_0 - T_k^s - T_k^B \le t_{ki}^{BP} \le \mathbf{FP}_0 - T_k^s, i = 1, 2, \text{ and } t_{k1}^{BP} + t_{k2}^{BP} \le 2(\mathbf{FP}_0 - T_k^s) - T_k^B\Big\} \tag{19d}$$

$$\mathcal{T}_5 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k1}^{BP} > \mathbf{FP}_0 - T_k^s, \text{ and } t_{k1}^{BP} - t_{k2}^{BP} \ge T_k^B\Big\} \tag{19e}$$

$$\mathcal{T}_6 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k2}^{BP} > \mathbf{FP}_0 - T_k^s, \text{ and } t_{k2}^{BP} - t_{k1}^{BP} \ge T_k^B\Big\} \tag{19f}$$

$$\mathcal{T}_7 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k1}^{BP} + t_{k2}^{BP} > 2(\mathbf{FP}_0 - T_k^s) - T_k^B, \text{ and } \big|t_{k2}^{BP} - t_{k1}^{BP}\big| < T_k^B\Big\}. \tag{19g}$$

Moreover, the pull strategy for single-blade robots and swap strategy for double-blade robots can be used to achieve **FP** calculated above.

Proof: See Appendix C. ■

Remark 3: It is interesting to point out that transfer cluster $\mathbb{C}_k$ could be considered as a special one-wafer capacity BPM with processing time $t_{k1}^{BP} = t_{k2}^{BP} = 2T_k$. Therefore, from (28), we can obtain the requirement for a feasible transfer cluster schedule as

$$4T_k \le \mathbf{FP}_0 - t_{k+1}^R - 2T_{k+1} - 2r_{k-1}T_{k-1}.$$

Remark 4: It is noted that when the long BPM processing time dominates $\mathbf{FP}_0$, there could exist some robot schedules that result in a smaller (average) fundamental period than those values calculated by (16) and (17). This is due to the fact that the *non one-wafer cycle* schedule could generate a smaller average cycle time. We can guarantee that the calculated **FP** by the first case in (16) and first two cases in (17) is the minimal one-wafer cycle time. However, for the second case in (16) and third case in (17), it is guaranteed that these values are the minimal one-wafer cycle times for pull and swap strategies. Discussion of the non one-wafer cycle production is out of the scope of this paper and readers can refer to Geismar *et al.* [33].

The BPM analysis can be integrated into the fundamental period computation algorithms discussed in the previous section. Suppose that there exist Q BPMs within the M-cluster tool, where $Q \le M - 1$, and we denote the BPM indexing set as $\mathcal{Q}$. We can calculate the fundamental period of the cluster tool with BPMs based on Proposition 3. Algorithm 3 describes such a modified fundamental period calculation.

Algorithm 3 : **FP** calculation of a cluster tool with BPMs.

```
Input : Cluster tool configuration and wafer flow $\mathbf{V}$
Output: Fundamental period $\mathbf{FP}$ for $\mathbf{V}$
Calculate $\mathbf{FP}_0$ assuming $t_{ij}^{BP} = 0, i \in \mathcal{Q}, j = 1, 2$ by Algorithm 1.
for $i \in \mathcal{Q}$ do
    Calculate $\mathbf{FP}_i$ for each BPM $BP_i$ using (16) or (17).
end
$\mathbf{FP} = \max_{i \in \mathcal{Q}}\{\mathbf{FP}_i\}$.
```

B. Robot Scheduling With BPMs

For robot scheduling with BPMs, we use the method that is described in Algorithm 2. We can incorporate the discussion in Appendix C into Algorithm 2. It is proper to schedule in the way such that BP_{k2} process ends right before action "R_k picking wafer from BP_{k2}" starts, and BP_{k1} process starts right after action "R_k placing wafer into BP_{k1}" ends. Then, in Algorithm 2, we can modify the following calculation for BPMs:

$$T_i^{\text{shift}} \leftarrow ST_{i-1}^s - 2T_i - t_{(i-1)2}^{BP}. \tag{20}$$

VII. Experimental Examples

We have successfully applied the methodology described in this paper to dozens of tools at Intel Corporation. The benefits include better throughput estimation, faster what-if analysis, and optimal scheduling solutions. Due to the page limit, we cannot discuss these advantages in details. In this section, we only demonstrate one example to show how to apply the proposed methodology in semiconductor manufacturing practice.

A. ALD/CVD Cluster Tool

Thin-film tools are widely used in semiconductor manufacturing to deposit metals onto silicon wafer surface using ei-

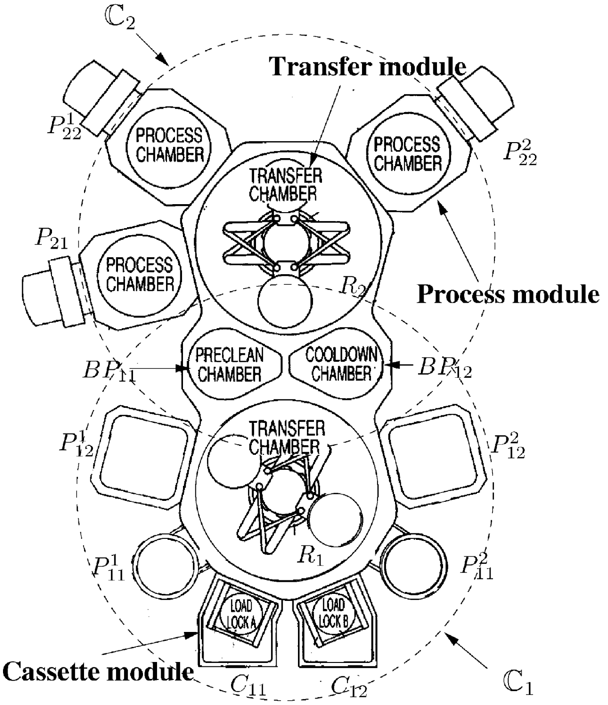


Fig. 8. A schematic layout of a CVD cluster tool.

TABLE I
PROCESS AND TRANSFER TIME OF THE CVD CLUSTER TOOL

Activities	Time variables	Values (s)
R_1 pick/place	T_1	10.9
R_2 pick/place	T_2	10.5
P_{11} processing	t_{11}	78.7
P_{12} processing	t_{12}	144.3
BP_{11} processing	t^{BP}_{11}	0
BP_{12} processing	t^{BP}_{12}	68.6
P_{21} processing	t_{21}	64.4
P_{22} processing	t_{22}	230.6

TABLE II
COMPUTATIONAL RESULTS FOR THE CVD CLUSTER TOOL BY ALGORITHM 1
("D" FOR DOUBLE-BLADE; "S" FOR SINGLE-BLADE)

$\mathbb{C}_i$	$\mathbf{S}_i$	R_i	N_i	$t^{d*}_{i(N_i+1)}$ (s)	t^{d*}_{i0}	$\mathbf{FP}^{d*}_i$ (s)
$\mathbb{C}_2$	–	D	2	–	0	125.8
$\mathbb{C}_1$	2	D	2	5	0	83.05

ther CVD, physical vapor deposition (PVD), sputter, atomic layer deposition (ALD), or electroplate processes. Fig. 8 shows a layout of an ALD/CVD cluster tool.[7] This is a two-cluster tool. The service cluster $\mathbb{C}_1$ includes a double-blade robot R_1, cassette C_{11} and C_{12}, and four process modules (chambers): parallel process modules P^1_{11} and P^2_{11}, and parallel process modules P^1_{12} and P^2_{12}. As discussed in Section IV-C, P^1_{11} and P^2_{11}, and P^1_{12} and P^2_{12} perform the same function, respectively. The processing cluster $\mathbb{C}_2$ includes a double-blade robot R_2 and three process modules (chambers). Two of these three process modules, P^1_{22} and P^2_{22}, are parallel modules. The interconnection BMs between $\mathbb{C}_1$ and $\mathbb{C}_2$ are BP_{11} and BP_{12}. All processing wafers on this cluster tool follow the visit route (the split arrows indicate the flow at parallel PMs):

$$C_{11} \xrightarrow{R_1} \left[\begin{array}{l} \longrightarrow P^1_{11} \xrightarrow{R_1} P^1_{12} \\ \longrightarrow P^2_{11} \xrightarrow{R_1} P^2_{12} \end{array}\right] \xrightarrow{R_1} BP_{11} \xrightarrow{R_2} P_{21} \longrightarrow$$

$$\xrightarrow{R_2} \left[\begin{array}{l} \longrightarrow P^1_{22} \\ \longrightarrow P^2_{22} \end{array}\right] \xrightarrow{R_2} BP_{12} \xrightarrow{R_1} C_{12}.$$

The processing time and robot transfer time for the CVD cluster tool are listed in Table I.

B. Maximum Throughput and Robot Schedule Results

The CVD cluster tool can be decomposed into two single-clusters, $\mathbb{C}_1$ and $\mathbb{C}_2$, as shown in Fig. 8. We can directly apply the decomposition technique discussed in Section V to this two-cluster tool.

Table II illustrates the maximum throughput calculation for the CVD cluster tool using the decoupled single-cluster approach (Algorithm 1). Here, we have to use (10) for parallel process modules in both clusters $\mathbb{C}_1$ and $\mathbb{C}_2$. For $\mathbb{C}_1$, the redundancy level is $l_{11} = l_{12} = 2$, and from (10)–(11) and Table I, we have

$$\mathbf{FP}^{d*}_1 = 2T_1 + \max\left\{\frac{t_{11} - 2T_1}{2}, \frac{t_{12} - 2T_1}{2}, t^{d*}_{10}, t^{d*}_{13}, 2(N_1 + 2 - 1)T_1\right\} = 83.05\text{ s}. \quad (21)$$

Similarly, for $\mathbb{C}_2$, we have

$$\mathbf{FP}^{d*}_2 = 2T_2 + \max\left\{\frac{t_{22} - 2T_2}{2}, t_{21}, t^{d*}_{20}, 2(N_2 + 1 - 1)T_2\right\} = 125.8\text{ s}. \quad (22)$$

We then can calculate the fundamental period $\mathbf{FP}_0$ assuming zero BPM processing times. Using Algorithm 1, we can calculate the $\mathbf{FP}_0$ of the two-cluster CVD cluster tool as

$$\mathbf{FP}_0 = \max\{\mathbf{FP}^{d*}_1, \mathbf{FP}^{d*}_2\} = 125.8\text{ s}.$$

For BPMs BP_{11} and BP_{12}, we can find that

$$\max\{t^{BP}_{11}, t^{BP}_{12}\} + T_1 + T_2 = 90 < \mathbf{FP}_0 = 125.8\text{ s}. \quad (23)$$

[7] Detailed information about such a type of cluster tool can be found at http://www.amat.com/products.

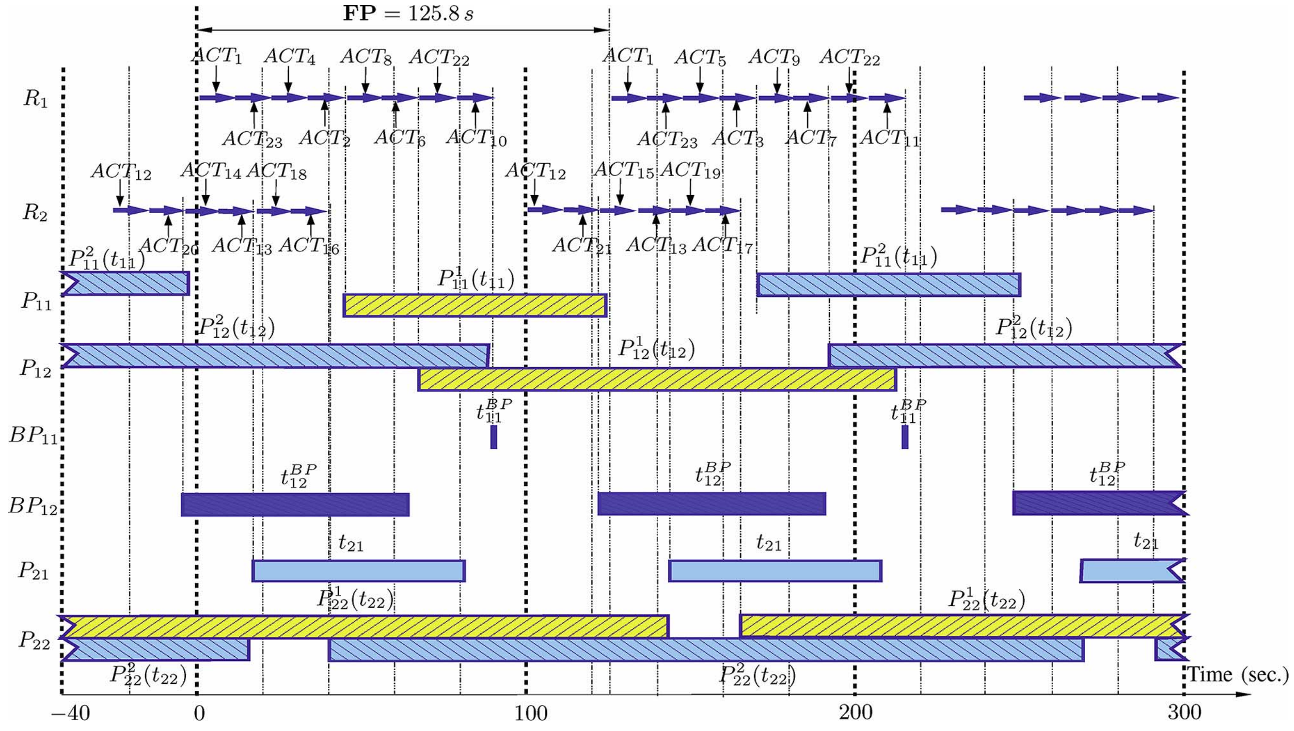


Fig. 9. Gantt chart of robot action (arrows) and processing sequences (rectangular blocks) of the CVD cluster tool. Parallel process modules are marked by different patterns with their corresponding symbols and processing time (in the brackets). BPMs are marked by dark colored blocks.

Therefore, by (17), BP_{11} and BP_{12} do not have any impact on the entire cluster tool throughput. Therefore, the fundamental period for the cluster tool is

$$\mathbf{FP} = \mathbf{FP}_0 = 125.8 \text{ s}. \tag{24}$$

If we increase BPM processing time to $125.8 - (T_1 + T_2) = 104.4$ s, then BPM starts dominating the cycle time of the cluster tool. This fact can be clearly seen in Gantt chart of CVD cluster tool shown in Fig. 9.

From Proposition 2, we know that the computed **FP** for the CVD cluster tool is achievable. To illustrate the optimal schedule, we label all robot actions as in Table III. Following Algorithm 2, we first find the optimal schedule according to the minimal cassette waiting time pull strategy for each decoupled cluster. Then, for the decomposed cluster of R_1, we have $ST_1^s = 65.4$ s as the starting time for ACT_{22}. With (20), we further obtain the relationship between the two decomposed clusters as

$$T_1^{\text{shift}} = ST_1^s - 2T_2 - t_{12}^{BP} = 65.4 - 2 \times 10.5 - 68.6 = -24.2 \text{ s}.$$

Therefore, we compute the robot schedules,[8] as shown in Table III and Fig. 9, which complies with the calculated **FP**.

It is also noted that in Fig. 9 a two-wafer cycle is shown repeatedly because there exist three sets of parallel process modules and their redundancy level are two, i.e., $\lambda = \text{LCM}(l_{11}, l_{12}, l_{21}) = \text{LCM}(2,2,2) = 2$. Therefore, the robot actions repeat every two-wafer cycle.

TABLE III
ACTION LABELS FOR THE CVD CLUSTER TOOL

ACT_i	Actions	Robot blade	Time (s)	Start time (s)
1	$C_{11} \to P_{11}$ pick	R_{11}/R_{12}	10.9	0/125.8
2	$C_{11} \to P_{11}^1$ place	R_{11}	10.9	32.7
3	$C_{11} \to P_{11}^2$ place	R_{12}	10.9	158.5
4	$P_{11}^1 \to P_{12}^1$ pick	R_{11}	10.9	21.8
5	$P_{11}^2 \to P_{12}^2$ pick	R_{12}	10.9	147.6
6	$P_{11}^1 \to P_{12}^1$ place	R_{11}	10.9	54.5
7	$P_{11}^2 \to P_{12}^2$ place	R_{12}	10.9	180.3
8	$P_{12}^1 \to BP_{11}$ pick	R_{11}	10.9	43.6
9	$P_{12}^2 \to BP_{11}$ pick	R_{12}	10.9	169.4
10	$P_{11}^1 \to BP_{11}$ place	R_{11}	10.9	76.3
11	$P_{11}^2 \to BP_{11}$ place	R_{12}	10.9	202.1
12	$BP_{11} \to P_{21}$ pick	R_{21}/R_{22}	10.5	$-24.2/101.6$
13	$BP_{11} \to P_{21}$ place	R_{21}/R_{22}	10.5	$7.3/133.1$
14	$P_{21} \to P_{22}^1$ pick	R_{21}	10.5	-3.2
15	$P_{21} \to P_{22}^2$ pick	R_{22}	10.5	122.6
16	$P_{21} \to P_{22}^1$ place	R_{21}	10.5	28.3
17	$P_{21} \to P_{22}^2$ place	R_{22}	10.5	154.1
18	$P_{22}^1 \to BP_{12}$ pick	R_{21}	10.5	17.8
19	$P_{22}^2 \to BP_{12}$ pick	R_{22}	10.5	143.6
20	$P_{22}^1 \to BP_{12}$ place	R_{21}	10.5	-13.7
21	$P_{22}^2 \to BP_{12}$ place	R_{22}	10.5	112.1
22	$BP_{12} \to C_{12}$ pick	R_{11}	10.9	65.4/191.2
23	$BP_{12} \to C_{12}$ place	R_{11}	10.9	10.9/136.7

We further use an alternative simulation based method [28] to verify the optimal scheduling for the CVD cluster tool. The

[8] The starting time for some robot actions (e.g., ACT_1) in Table III are listed twice because of the two-wafer cycle due to the existence of several two-parallel PMs.

simulation gives the same results. The production at one Intel Corporation fab achieved a 28.6 wafers per hour throughput (125.8 s cycle time per wafer) at the steady state. The production results further validate the analytical and simulation studies.

VIII. Conclusion and Future Work

In this paper, we presented a decomposition method to analyze the steady-state throughput and robot scheduling of a multicluster tool for semiconductor manufacturing. We considered a production case in which all processing wafers follow the same visit route. We first extended the existing single-cluster scheduling results to a robot minimal cassette waiting time pull and swap strategies for single- and double-blade robots. Based on these extensions, we discussed the lower-bound cycle time of multicluster tools using a decomposition method. We then presented optimality conditions under which such a lower-bound cycle time is feasible. Algorithms to compute the maximum throughput and to search a feasible optimal schedule of multicluster tools were proposed and analyzed. The impact of the combined BPMs on cluster tool throughput and scheduling was also analyzed. The proposed analytical and computational approach provided an efficient method to study the throughput and scheduling of multicluster tools. An application example of a CVD cluster tool at Intel Corporation has been used to illustrate the proposed decomposition methods.

There are several future research directions. In semiconductor manufacturing, the processing times at one or several process modules could vary due to the incoming film thickness variations and process shifts and drifts. A natural extension is cyclic scheduling and analysis of a multicluster tool with random processing times. A preliminary study for such a problem has been reported in [34] using a network flow model. One interesting problem is to design BM capacity around bottleneck clusters with random processing time and to, hence, reduce throughput variations. The increasing demands of in-line metrology and dynamic manufacturing would require reentrant and mixed wafer visit patterns in cluster tools. Considering the optimal robot schedules under such requirements is also an interesting problem in future research.

Appendix A
Proof of Proposition 1

By the definition of cassette waiting time t^R, it is straightforward to obtain $t^R = 0$ if robot R is double-blade ($r = 1$) and using the minimal cassette waiting time swap strategy.

If R is a single-blade robot ($r = 2$) and cluster is running in robot-bound region, i.e., $t^{\max} \leq 2(N-1)T$, then $t^R = 2T$ and (9) still holds since $t^{\min} \leq t^{\max} \leq 2(N-1)T$.

Now, consider the case $t^{\max} > 2(N-1)T$ and the cluster is running in a process-bound region. In such a case, the total robot idle time is $t_{\text{idle}} = t^{\max} - 2(N-1)T > 0$. By the minimal cassette waiting time pull strategy for robot R, the maximal robot idle time at the PM with the minimal processing time $t^{\min}$ is

$$\begin{aligned} t_{\text{idle}}^{PM} &= \min\{\delta t, t_{\text{idle}}\} \\ &= \min\{t^{\max} - t^{\min}, t^{\max} - 2(N-1)T\}. \end{aligned}$$

Moreover, the total robot idle time t_{idle} satisfies (Fig. 4)

$$t_{\text{idle}} = (t_{\min}^R - 2T) + t_{\text{idle}}^{PM} \tag{25}$$

and therefore

$$\begin{aligned} t_{\min}^R &= t_{\text{idle}} - t_{\text{idle}}^{PM} + 2T \\ &= t^{\max} - 2(N-1)T - \\ &\quad \min\{t^{\max} - t^{\min}, t^{\max} - 2(N-1)T\} + 2T \\ &= \max\{t^{\min} - 2(N-1)T, 0\} + 2T \end{aligned}$$

which proves Proposition 1.

Appendix B
Proof of Proposition 2

The optimality conditions come directly from the analysis and computing algorithms of $\mathbf{FP}$. When we compute the lower-bound value $\mathbf{FP}_k^{d*}$, we use a minimal interaction time between two adjacent clusters for t_{k0}^{d*} and $t_{k(N_k+1)}^{d*}$. The important factor for such a minimal time realization is the robot cassette waiting time t_k^R. We know that if all robots are double-blade, then $t_k^R = 0$ and any robots pick/place actions for computing $\mathbf{FP}$ can be realized. For single-blade robot R_k, if $\mathbb{C}_k$ is a transfer cluster, $\mathbf{FP}_k^{d*}$ by (11) captures the exact waiting time at $P_{k(N_k+1)}^*$ and C_k^*; if $\mathbb{C}_k$ is not a transfer cluster, then we can utilize the results from the analysis given in Section VI for BPM by considering zero BPM processing time, i.e., $t_{(k-1)1}^{BP} = t_{(k-1)2}^{BP} = 0$. Here, the conditions to maintain $\mathbf{FP}_0$ in BPM analysis need to be enforced. If $\mathbf{S}_{k-1} = 1$, from condition (28), we have

$$\begin{aligned} 0 \leq \mathbf{FP} - 2T_{k-1}^s - T_{k-1}^B = \mathbf{FP} - 2(T_{k-1} + T_k) - \\ \left[t_k^R + 2(r_{k-1} - 1)T_{k-1}\right] \end{aligned}$$

therefore

$$t_k^R \leq \mathbf{FP} - 2r_{k-1}T_{k-1} - 2T_k\,.$$

If $\mathbf{S}_{k-1} = 2$, from conditions (19) and (17), we have

$$\begin{aligned} 0 \leq 2(\mathbf{FP} - T_{k-1}^s) - T_{k-1}^B = 2\left[\mathbf{FP} - (T_{k-1} + T_k)\right] - \\ \left[t_k^R + 2(r_{k-1} - 1)T_{k-1}\right] \end{aligned}$$

thus

$$t_k^R \leq 2\mathbf{FP} - 2r_{k-1}T_{k-1} - 2T_k\,.$$

This completes the proof of Proposition 2.

Appendix C
Proof of Proposition 3

To analyze the robot action constraints on BPM BP_k, we denote the robots R_k and R_{k+1} actions during Kth cycle as follows (Fig. 6): $ACT_{k_1 I}^{(K)}$ and $ACT_{k_2 O}^{(K)}$ for robot R_k's placing wafer into BP_{k1} and picking wafer from BP_{k2}, respectively; $ACT_{k_2 I}^{(K)}$ and $ACT_{k_1 O}^{(K)}$ for robot R_{k+1}'s placing wafer into BP_{k2} and picking wafer from BP_{k1}, respectively.

Using the decomposition approach, the pull strategy for single-blade robots and the swap strategy for double-blade robots, the following robot action sequence constraints must be satisfied.

Constraint 1: BPM operation constraints under pull and swap strategies.

1) $ACT_{k_1I}^{(K+1)}$ starts after $ACT_{k_1O}^{(K)}$ is finished.
2) $ACT_{k_2I}^{(K)}$ starts after $ACT_{k_2O}^{(K-1)}$ is finished.
3) BP_{k1} process starts right after $ACT_{k_1I}^{(K)}$ is finished and $ACT_{k_1O}^{(K)}$ starts after BP_{k1} process is finished.
4) BP_{k2} process starts right after $ACT_{k_2I}^{(K)}$ is finished and $ACT_{k_2O}^{(K)}$ starts after BP_{k1} process is finished.
5) $ACT_{k_1I}^{(K)}$ follows after $ACT_{k_2O}^{(K-1)}$ is finished.
6) $ACT_{k_2I}^{(K)}$ follows after $ACT_{k_1O}^{(K-1)}$ is finished.

The first four constraints are required for feasible schedules of the BPMs. Constraints 1.5 and 1.6 are from the action sequences of the pull strategy for single-blade robots and swap strategy for double-blade robots. It is also noted that if the pull and swap strategies are strictly enforced, constraints 1.5 and 1.6 can be written as follows:

$$t(ACT_{k_1I}^{(K)}) - t(ACT_{k_2O}^{(K-1)}) = 2(r_k - 1)T_k \qquad (26)$$

$$t(ACT_{k_2I}^{(K)}) - t(ACT_{k_1O}^{(K-1)}) = t_{k+1}^{R} \qquad (27)$$

where $t(ACT)$ denotes the starting time of ACT.

In the following, we first analyze the robot R_k's and R_{k+1}'s actions on BP_{k1} and BP_{k2} and discuss the conditions on BPM processing time t_{k1}^{BP} and t_{k2}^{BP} under which the calculated fundamental period $\mathbf{FP}_0$ is still valid. Then,we extend such an analysis to the cases when BPMs dominate $\mathbf{FP}_0$ and give the corresponding formula to calculate $\mathbf{FP}$ under such cases. We begin with one-wafer capacity BPMs, and then discuss two-wafer capacity BPMs.

One-Wafer Capacity ($\mathbf{S}_k = 1$) *BPM:* In this case, the incoming and outgoing wafers between $\mathbb{C}_k$ and $\mathbb{C}_{k+1}$ share the same BM, namely, BP_{k1} and BP_{k2} in Fig. 6 is physically the same device. Without loss of generality, we consider the Kth robot action cycle on BP_{k1} and BP_{k2} at the steady state.

Case 1: Non-BPM Bound: If the BPM processing time $(t_{k1}^{BP} + t_{k2}^{BP})$ is small, $\mathbf{FP}_0$ can still be maintained and BPM does not have to be a constraint. Fig. 10 shows the Gantt chart of the robot actions under which the cluster tool runs at the fundamental period $\mathbf{FP}_0$.

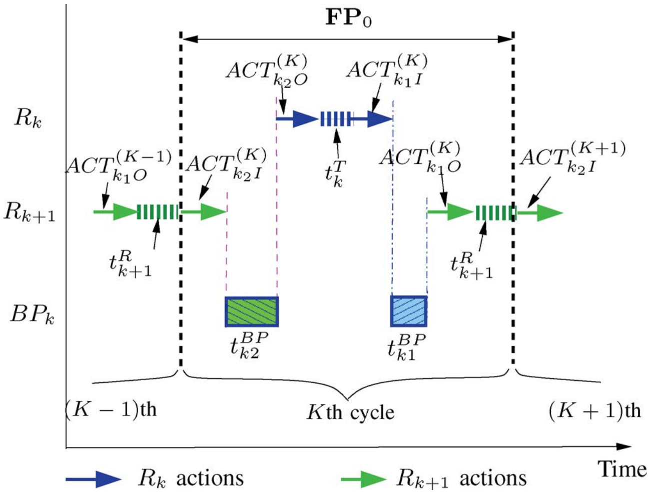


Fig. 10. Gantt chart of robot moving actions for a one-wafer capacity ($\mathbf{S}_k = 1$) BPM between clusters $\mathbb{C}_k$ and $\mathbb{C}_{k+1}$.

From Fig. 10, it is observed that the requirement for such a non-BPM bound case is

$$2T_{k+1} + 2T_k + t_k^T + t_{k+1}^R + t_{k1}^{BP} + t_{k2}^{BP} \le \mathbf{FP}_0$$

namely

$$t_{k1}^{BP} + t_{k2}^{BP} \le \mathbf{FP}_0 - 2T_k^s - T_k^B. \qquad (28)$$

Case 2: BPM Bound: If BPM processing time condition (28) is not satisfied, then $\mathbf{FP}_0$ cannot be maintained by the pull and swap strategies. We can obtain that under such a BPM bound case the fundamental period of the cluster tool is given by

$$\mathbf{FP} = t_{k1}^{BP} + t_{k2}^{BP} + T_k^B + 2T_k^s.$$

Thus, we can summarize the discussion above as (16).

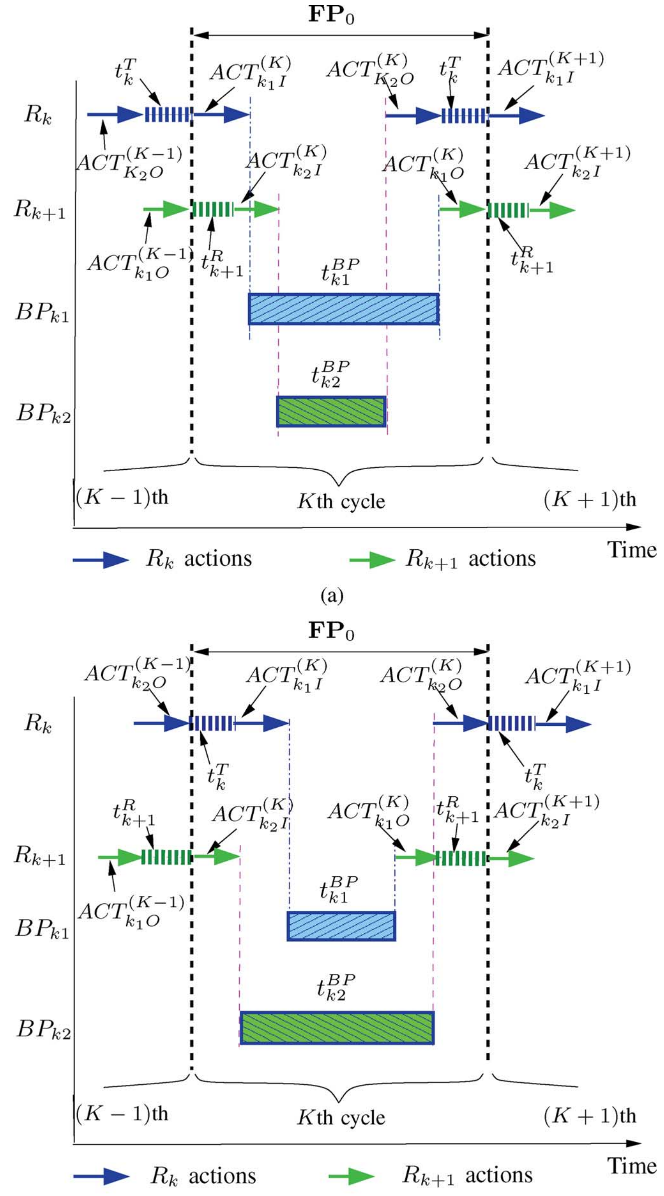
Two-Wafer Capacity ($\mathbf{S}_k = 2$) *BPM:*

Case 1: Non-BPM Bound: Fig. 11 shows the two cases under which the cluster tool runs at the fundamental period $\mathbf{FP}_0$.

Depending on the processing times t_{k1}^{BP} and t_{k2}^{BP} and cluster configurations, we can find the conditions under which the cluster tool can achieve $\mathbf{FP}_0$.

1) $t_{ki}^{BP} \le \mathbf{FP}_0 - T_k^s - T_k^B$, $i = 1, 2$. In this case, the BPM processing times are relatively small comparing with $\mathbf{FP}_0$. The cluster tool could achieve $\mathbf{FP}_0$ through a robot scheduling algorithm discussed above.
2) $t_{k2}^{BP} \le \mathbf{FP}_0 - T_k^s - T_k^B$ *and* $\mathbf{FP}_0 - T_k^s - T_k^B \le t_{k1}^{BP} \le \mathbf{FP}_0 - T_k^s$. Fig. 11(a) shows the Gantt chart for an extreme case ($t_{k1}^{BP} = \mathbf{FP}_0 - T_k^s$), where the feasible robot schedule can achieve $\mathbf{FP}_0$. In this case, process BP_{k1} dominates BP_{k2} and the maximum allowable processing time for BP_{k1} is $\mathbf{FP}_0 - T^s$. Under such a situation, Constraint 1.1 becomes tight and T_k^B (minimal robot waiting time t_{k+1}^R of $\mathbb{C}_{k+1}$ plus robot transferring time $2(r_k - 1)T_k$ of $\mathbb{C}_k$) can be fit into BP_{k2} processing cycle [Fig. 11(a)].
3) $t_{k1}^{BP} \le \mathbf{FP}_0 - T_k^s - T_k^B$ *and* $\mathbf{FP}_0 - T_k^s - T_k^B \le t_{k2}^{BP} \le \mathbf{FP}_0 - T_k^s$. Fig. 11(b) shows the Gantt chart for an extreme case ($t_{k2}^{BP} = \mathbf{FP}_0 - T_k^s$), where the feasible robot schedule can achieve $\mathbf{FP}_0$. Similar to the previous case, process BP_{k2} instead dominates BP_{k1} and the maximum allowable processing time for BP_{k2} is $\mathbf{FP}_0 - T^s$. In such a situation, Constraint 1.2 becomes tight and T_k^B falls into BP_{k1} processing cycle [Fig. 11(b)].
4) $\mathbf{FP}_0 - T_k^s - T_k^B \le t_{ki}^{BP} \le \mathbf{FP}_0 - T_k^s$, $i = 1, 2$, *and* $t_{k1}^{BP} + t_{k2}^{BP} \le 2(\mathbf{FP}_0 - T_k^s) - T_k^B$. In this case, the robot actions fall between the two extreme cases above. We can coordinate robots R_k and R_{k+1} such that T_k^B falls into both BP_{k1} and BP_{k2} processing period within $\mathbf{FP}_0$.

Graphically, we can consider the BPM processing times in the t_{k1}^{BP}-t_{k2}^{BP} plane and summarize the four cases above into four regions $\mathcal{T}_1$-$\mathcal{T}_4$ defined by (19a)–(19d). When $(t_{k1}^{BP}, t_{k2}^{BP}) \in \bigcup_{i=1}^{4} \mathcal{T}_i$ (as shown in regions 1–4 in Fig. 7), the cluster tool can maintain $\mathbf{FP}_0$.



(a)

(b)

Fig. 11. Gantt chart of robot moving action for a two-wafer capacity ($\mathbf{S}_k = 2$) BPM between clusters $\mathbb{C}_k$ and $\mathbb{C}_{k+1}$. (a) BP_{k1} domination case. (b) BP_{k2} domination case.

Case 2: BPM Bound: With the increase of BPM processing time, BPMs eventually could become bottleneck of the cluster tool. With the analysis above, we can carry out a similar analysis when one (or more) of the BPMs dominates the cluster tool cycle time.

1) BP_{k1} or BP_{k2} domination. If BP_{k1} process dominates, $t_{k1}^{BP} > \mathbf{FP}_0 - T_k^s$. Constraints 1.3 are bounded and Gantt chart in Fig. 11(a) can be used to calculate the fundamental period of the cluster tool. It is noted that **FP** can be calculated as

$$\mathbf{FP} = t_{k1}^{BP} + T_k^s.$$

In this case, it also requires that BP_{k2} satisfies

$$t_{k2}^{BP} \le t_{k1}^{BP} - T_k^B.$$

In the t_{k1}^{BP}-t_{k2}^{BP} plane (Fig. 7), this corresponds to region $\mathcal{T}_5$ defined by (19e).

Similar results could be found if BP_{k2} process dominates

$$\mathbf{FP} = t_{k2}^{BP} + T_k^s, \text{ and } t_{k1}^{BP} \le t_{k2}^{BP} - T_k^B.$$

In the t_{k1}^{BP}-t_{k2}^{BP} plane (Fig. 7), this corresponds to region $\mathcal{T}_6$ defined by (19f).

2) If $|t_{k2}^{BP} - t_{k1}^{BP}| < T_k^B$ and $t_{k1}^{BP} + t_{k1}^{BP} > 2(\mathbf{FP}_0 - T_k^s) - T_k^B$, both BPMs processing times are large and they are within a range of T_k^B. In such a case, we can derive that the fundamental period should be

$$\mathbf{FP} = \frac{t_{k1}^{BP} + t_{k2}^{BP} + T_k^B}{2} + T_k^s.$$

This result could be obtained by combining the Gantt charts in Fig. 11(a) and (b). In the t_{k1}^{BP}-t_{k2}^{BP} plane (Fig. 7), this corresponds to region $\mathcal{T}_7$ defined by (19g).

With the discussion above, we can summarize the **FP** calculation as (17) and also represent in various regions in the t_{k1}^{BP}-t_{k2}^{BP} plane, as shown in Fig. 7. This completes the proof of Proposition 3.

Acknowledgment

The authors thank the Associate Editor and three anonymous reviewers for their constructive feedback and suggestions. They are also grateful to Prof. W.-K. (Victor) Chan at Rensselaer Polytechnic Institute for various helpful discussions and suggestions.

References

[1] "SEMI E21 – Cluster tool module interface: mechanical interface and wafer transport standard," Semiconductor Equipment and Materials International (SEMI), 1996. [Online]. Available: http://www.semi.org

[2] D. Jevtic, "Method and apparatus for managing scheduling a multiple cluster tool," Eur. Patent 1,132,792 (A2), Dec. 2001.

[3] M. Dawande, H. Geismar, S. Sethi, and C. Sriskandarajah, "Sequencing and scheduling in robotic cells: Recent developments," *J. Scheduling*, vol. 8, no. 5, pp. 387–426, 2005.

[4] M. Pinedo*, Scheduling : Theory, Algorithms, and Systems*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[5] T. Perkinson, P. McLarty, R. Gyurcsik, and R. Cavin, "Single-wafer cluster tool performance: An analysis of throughput," *IEEE Trans. Semiconduct. Manufact.*, vol. 7, no. 3, pp. 369–373, 1994.

[6] 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. Semiconduct. Manufact.*, vol. 10, no. 4, pp. 418–424, 1997.

[7] Y. Crama and J. van de Klundert, "Cyclic scheduling of identical parts in a robotic cell," *Oper. Res.*, vol. 45, no. 6, pp. 952–965, 1997.

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

[9] I. Drobouchevitch, S. Sethi, and C. Sriskandarajah, "Scheduling dual gripper robotic cells: One-unit cycles," *Eur. J. Oper. Res.*, vol. 171, no. 2, pp. 598–631, 2006.

[10] Q. Su and F. Chen, "Optimal sequencing of double-gripper granty robot moves in tightly-coupled serial production systems," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 22–30, 1996.

[11] Y. Crama, V. Kats, J. van de Klundert, and E. Levner, "Cyclic scheduling of robotic flowshops," *Ann. Oper. Res.*, vol. 96, no. 1, pp. 97–124, 2000.

[12] S. Sethi, J. Sidney, and C. Sriskandarajah, "Scheduling in dual gripper robotic cells for productivity gains," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 324–341, Jun. 2001.

[13] M. Zhou and M. Jeng, "Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing: A Petri net approach," *IEEE Trans. Semiconduct. Manufact.*, vol. 11, pp. 333–357, 1998.

[14] R. Srinivasan, "Modeling and performance analysis of cluster tools using Petri nets," *IEEE Trans. Semiconduct. Manufact.*, vol. 11, no. 3, pp. 394–403, Aug. 1998.

[15] W. Zuberek, "Timed Petri nets in modeling and analysis of cluster tools," *IEEE Trans. Robot. Automat.*, vol. 17, no. 5, pp. 562–575, Oct. 2001.

[16] N. Wu and M. C. Zhou, "Schedulability and scheduling of dual-arm cluster tools with residency time constraints based on Petri net," in *Proc. IEEE Conf. Autom. Sci. Eng.*, Shanghai, China, 2006, pp. 85–90.

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

[18] S. Rostami and B. Hamidzadeh, "Optimal scheduling techniques for cluster tools with process-module and transport-module residency constraints," *IEEE Trans. Semiconduct. Manufact.*, vol. 15, no. 3, pp. 341–349, Aug. 2002.

[19] H. T. LeBaron and R. A. Hendrickson, "Using emulation to validate a cluster tool simulation model," in *Proc. Winter Simulation Conf.*, Orlando, FL, 2000, pp. 1417–1422.

[20] J. Kim, T. Lee, H. Lee, and D. Park, "Scheduling analysis of time-constrained dual-armed cluster tools," *IEEE Trans. Semiconduct. Manufact.*, vol. 16, no. 3, pp. 521–534, Aug. 2003.

[21] Y. Joo and T. Lee, "Virtual control – A virtual cluster tool for testing and verifying a cluster tool controller and a scheduler," *IEEE Robot. Automat. Mag.*, vol. 11, no. 3, pp. 33–49, 2004.

[22] D. A. Nehme and N. G. Pierce, "Evaluating the throughput of cluster tools using event-graph simulations," in *Proc. IEEE/SEMI Adv. Semiconduct. Manufact. Conf.*, Cambridge, MA, 1994, pp. 189–192.

[23] D. Pederson and C. Trout, "Demonstrated benefits of cluster tool simulation," in *Proc. Int. Conf. Modeling Anal. Semiconduct. Manufact.*, Tempe, AZ, 2002, pp. 84–89.

[24] S. Ding and J. Yi, "An event graph based simulation and scheduling analysis of multi-cluster tools," in *Proc. Winter Simulation Conf.*, Washington, DC, 2004, pp. 1915–1924.

[25] T. Perkinson, R. Gyurcsik, and P. McLarty, "Single-wafer cluster tool performance: An analysis of the effects of redundant chambers and revisitation sequences on throughput," *IEEE Trans. Semiconduct. Manufact.*, vol. 9, no. 3, pp. 384–400, Aug. 1996.

[26] N. Geismar, M. Dawande, and C. Sriskandarajah, "Robotic cells with parallel machines: Throughput maximization in constant travel-time cells," *J. Scheduling*, vol. 7, no. 5, pp. 375–395, 2004.

[27] J. Herrmann, N. Chandrasekaran, B. Conaghan, M. Nguyen, G. Rubloff, and R. Zhi, "Evaluating the impact of process changes on cluster tool performance," *IEEE Trans. Semiconduct. Manufact.*, vol. 13, no. 2, pp. 181–192, May 2000.

[28] S. Ding, J. Yi, and M. T. Zhang, "Multi-cluster tools scheduling: An integrated event graph and network model approach," *IEEE Trans. Semiconduct. Manufact.*, vol. 19, no. 3, pp. 339–351, 2006.

[29] N. Geismar, C. Sriskandarajah, and N. Ramanan, "Increasing throughput for robotic cells with parallel machines and multiple robots," *IEEE Trans. Automat. Sci. Eng.*, vol. 1, no. 1, pp. 84–89, Jul. 2004.

[30] 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," U.S. Patent 6,224,638, May 2001.

[31] J. Yi, S. Ding, and D. Song, "Steady-state throughput and scheduling analysis of multi-cluster tools for semiconductor manufacturing: An decomposition approach," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, 2005, pp. 293–299.

[32] J. Yi, S. Ding, D. Song, and M. T. Zhang, "Scheduling analysis of cluster tools with buffer/process modules," in *Proc. IEEE Int. Conf. Robot. Automat.*, Rome, Italy, 2007, pp. 985–990.

[33] N. Geismar, M. Dawande, and C. Sriskandarajah, "Approximation algorithms for k-unit cyclic solutions in robotic cells," *Eur. J. Oper. Res.*, vol. 162, no. 2, pp. 291–309, 2005.

[34] S. Ding, J. Yi, M. T. Zhang, and R. Akhavan-Tabatabaei, "Performance evaluation and schedule optimization of multi-cluster tools with stochastic process times," in *Proc. IEEE Conf. Automat. Sci. Eng.*, Shanghai, China, 2006, pp. 100–105.

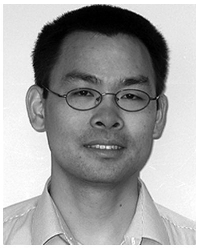
Jingang Yi (S'99–M'02) received the B.S. degree in electrical engineering from the Zhejiang University, Hangzhou, China, in 1993, the M.Eng. degree in precision instruments from Tsinghua University, Beijing, China, in 1996, the M.A. degree in mathematics, and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, in 2001 and 2002, respectively.

He is currently an Assistant Professor in Mechanical Engineering at San Diego State University. From May 2002 to January 2005, he was with Lam Research Corporation, Fremont, CA, as a member of Technical Staff. From January 2005 to December 2006, he was with the Department of Mechanical Engineering, Texas A&M University, as a Visiting Assistant Professor. His research interests include intelligent and autonomous systems, dynamic systems and control, intelligent sensing and actuation systems, mechatronics, automation science and engineering with applications to semiconductor manufacturing and intelligent transportation systems.

Dr. Yi is a member of American Society of Mechanical Engineering (ASME). He was the recipient of the Kayamori Best Paper Award of the 2005 IEEE Conference on Robotics and Automation (ICRA).

Shengwei Ding received the B.S. and M.S. degrees in electrical engineering from Zhejiang University, Zhejiang, China, in 1996 and 1999, respectively, and the Ph.D. degree in industrial engineering and operation research from the University of California, Berkeley, in 2004.

He is currently with the Department of Industrial Engineering and Operations Research, University of California, Berkeley. His research interests are queueing models, simulation, scheduling, production management, and semiconductor manufacturing.



Dezhen Song (S'02–M'04) received the Ph.D. degree in engineering from the University of California, Berkeley, in 2004.

Currently, he is an Assistant Professor with Texas A&M University, College Station. His research area is networked robotics, computer vision, optimization, and stochastic modeling.

Dr. Song received the Kayamori Best Paper Award at the 2005 IEEE International Conference on Robotics and Automation, with J. Yi and S. Ding. He received the NSF Faculty Early Career Development (CAREER) Award in 2007.

Mike Tao Zhang (S'98–M'01–SM'05) received the M.S. and Ph.D. degrees from the Department of Industrial Engineering and Operations Research, in 2000 and 2001, respectively, as well as the Management of Technology Certificate in 2000 from the Haas School of Business and the College of Engineering, University of California, Berkeley.

He is currently a Senior Manager of Systems Automation and Industrial Engineering at Spansion Inc., Sunnyvale, CA. He has been a Senior Engineer, a Group Leader, a Department Manager, and a Staff Engineer at various Intel sites. He was awarded three patents and published over 50 papers and four books/book chapters. His research interests are industrial automation, manufacturing systems, operations research/management, and supply chain management.

Dr. Zhang is a Member of the Honor Society of Phi Kappa Phi, and also a Senior Member of the Institute of Industrial Engineers (IIE). He is Co-Chair of the IEEE Robotics and Automation Society Technical Committee on Semiconductor Manufacturing Automation. He is an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and a Guest Editor of *Assembly Automation* and the IEEE ROBOTICS AND AUTOMATION MAGAZINE. He is Program Chair of the 2007 IEEE Conference on Automation Science and Engineering. He is also the recipient of the Intel ATM Achievement Award and the IIE Outstanding Young Industrial Engineer Award. He is listed in *Marquis Who's Who in the World*.