Optimal Preventive Maintenance Scheduling in Semiconductor Manufacturing

Xiaodong Yao, Emmanuel Fernández-Gaucherand, Michael C. Fu, and Steven I. Marcus

X. Yao, M. Fu, and S. Marcus are with the Institute for Systems Research, University of Maryland, College Park, MD 20742-1815 (e-mail: {xdyao, mfu, marcus}@isr.umd.edu).

E. Fernández-Gaucherand is with the Dept. of Electrical and Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH 45221-0030 (e-mail: emmanuel@ececs.uc.edu).

This work was supported in part by the Semiconductor Research Corporation and International SEMATECH under Grants 97-FJ-491 and NJ 877. Michael Fu and Steven Marcus were also supported in part by the National Science Foundation under Grants DMI-9713720 and DMI-9988867, and by the Air Force Office of Scientific Research under Grant F496200110161. A preliminary version of this work was presented at the IEEE Conference on Control Applications in 2001.

Abstract

Preventive Maintenance (PM) scheduling is a very challenging task in semiconductor manufacturing, due to the complexity of highly integrated fab tools and systems, the interdependence between PM tasks, and the balancing of Work-In-Process (WIP) with demand/throughput requirements. In this paper, we propose a two-level hierarchical modeling framework. At the higher level is a model for long-term planning, and at the lower level is a model for short-term PM scheduling. Solving the lower level problem is the focus of this paper. We develop mixed integer programming (MIP) models for scheduling all due PM tasks for a group of tools, over a planning horizon. Interdependence among different PM tasks, production planning data such as projected WIP levels, manpower constraints, and associated PM time windows and costs, are incorporated in the model. Results of a simulation study comparing the performance of the model-based PM schedule with that of a baseline reference schedule are also presented.

Index Terms

preventive maintenance, scheduling, mixed integer programming, time-window policies, cluster tools.

I. Introduction

The reliability of equipment in semiconductor manufacturing fabs has become an important issue in yield improvement, cost reduction, and cycle time reduction. The fabrication equipment is extremely sophisticated and costly, requiring extensive calibration and preventive maintenance (PM). A good PM plan can increase equipment availability by trading off between “planned” unproductive downtime (due to PM) and the risk of much costlier “unplanned” downtime (due to equipment failure), which can cause major disruptions in the manufacturing process. Thus, in order to maximize the profits from fab operations, PM tasks have to be scheduled carefully and comprehensively.

However, PM scheduling in semiconductor fabs has long been seen as a very hard problem [1], [2]. Two distinguishing features of semiconductor fabs make PM scheduling a particularly challenging task. First, a semiconductor fab is a highly integrated system that involves many different types of equipment and up to several hundred processing steps for each wafer. These equipment (also called tool) are highly coupled by re-entrant wafers and processing steps. A PM schedule on one tool can have significant impact on upstream or downstream tools if maintenance

is scheduled poorly. Second, many new advanced techniques have been deployed in fabs, such as the wide use of cluster tools. A cluster tool is a highly integrated machine that is composed of several chambers and robots, where different PM tasks on different chambers have to be coordinated carefully in order to maximize the availability (therefore throughput) of the entire tool.

For example, even the PM scheduling for a work center, in isolation, that consists of a group of cluster tools turns out to be quite complicated, as each cluster tool has several chambers, and each chamber has several different PM tasks that have to be performed. To improve the availability of the entire tool requires coordination of PM tasks in different chambers, because the entire tool's availability is dependent on the status of each chamber. In addition, fab production data such as Work-In-Process (WIP) should be considered in PM scheduling. For instance, PM tasks should be avoided if possible during periods when a significant amount of work is expected to arrive soon. It would be wise to "pull" or "push" a planned PM task beyond a certain period under such circumstances. Hence, PM tasks should be scheduled by looking ahead at both the effect from WIP and the impact on WIP. In addition, it may be advantageous sometimes to consolidate PM tasks, e.g., doing one task "early" when a tool or chamber is brought down for another task. Costs for supplies and lost production, as well as technician availability constraints, should also be accounted for.

It is obvious that the uncertain (stochastic) nature of WIP and tool failures, and the interdependence of PM tasks in fabs, require new models to be developed to deal with these complicated situations. Unfortunately, there do not appear to be such models readily applicable to PM scheduling problems. On the other hand, there are enormous amounts of data in the fab databases readily available to modelers and planners; yet most of this potential goes unutilized.

We argue that a decomposition approach can be employed for the problem of optimal PM scheduling. Specifically, a two-level hierarchical modeling framework is proposed. At the higher level, a *planning* model captures both the failure processes and the demand pattern, and can be used to derive optimal PM planning policies for the long-term horizon. At the lower level, a scheduling model, accepting as one input optimal policies from the higher level, takes into consideration the interdependence among PM tasks and resources constraints (e.g., headcount of maintenance technicians), and is used to determine the optimal time to do PMs for short-term operations. Following standard practice in industry, PM tasks are scheduled within a given

nominal window, e.g., 14 days after the last PM, plus or minus 1 day. The scheduling model is formulated as a mixed integer programming (MIP) model, and is the main focus of this paper.

The work presented here is part of a comprehensive effort at developing models, algorithms and software tools for PM scheduling. Significant interaction with many semiconductor manufacturing companies has taken place to date, primarily within the joint Semiconductor Research Corporation (SRC) and International SEMATECH (ISMT) "Factory Operations Research Center" (FORCe) program.

The main contributions of this paper are two-fold. First, we propose a systematic approach to the PM planning and scheduling problem in semiconductor manufacturing, i.e., the two-level hierarchical modeling framework, which is applicable to real situations in fabs. Second, we develop a mixed integer programming model for optimal PM scheduling in fabs. These solutions are applicable for all tool groups in a fab, but higher impact is to be expected when purely ad hoc scheduling may be too complex to handle, e.g., for cluster tools. In general, given their optimization base, our solutions can be a significant aid for (human) decision makers to rule out errors and oversights.

The remainder of the paper is organized as follows. Section II contains a brief literature review on PM models and scheduling. Section III introduces the hierarchical PM planning and scheduling framework. The MIP model for the PM scheduling problem is developed and discussed in section IV, with more technical details provided in the Appendix. The simulation study is contained in Section V. Finally we provide some concluding remarks.

II. Literature Review

Preventive maintenance theory has a well-established body of literature. Many maintenance models have been developed and applied in manufacturing systems. The survey papers [3], [4], [5], [6] provide a wide range of models describing the degrading process of equipment, cost structure and admissible maintenance actions. In general, the underlying stochastic processes can be modeled as a Markov chain or semi-Markov process. Normally, the cost structure could include inspection cost, preventive replacement cost, failure replacement cost, repair cost, and operation cost. One important result about optimal policies derived from these models is that they are often of the "*control limit*" form; for a general treatment, see [7]. For example, the well-known age-dependent PM policy is such a *control limit* policy, i.e., if machine's age is

beyond a *limit*, then it is optimal to do PM; otherwise do not perform PM.

Along with the development of maintenance models for single-unit systems, some work has been directed to multi-component (multi-unit) maintenance models, where several machines are stochastically or economically dependent on each other; see the survey paper [5] and the references therein. Most of these efforts have been focused on group/block or opportunistic maintenance models that make use of economies of scale to perform preventive replacement upon the failure of one unit, or on the investigation of the effect of repairmen/spare parts inventory on maintenance policies.

Traditionally, PM modeling has concentrated on utilizing data solely on the reliability of individual machines. However, this approach is not well suited for interdependent systems, such as modern semiconductor manufacturing fabs, which are characterized by high interdependence among different tools. Traditional methods have ignored the fact that each tool is only a part of the whole production system, and so the entire state of the system, such as the operating status of upstream or downstream tools, as well as buffer levels, has significant impact on PM scheduling for that tool and should be considered in order to achieve maximal overall equipment effectiveness.

Van der Duyn Schouten and Vanneste [8] investigate an integrated maintenance-production problem, in which the preventive maintenance policy is based not only on the information about the age of the device, but also on the level of the downstream buffer. Meller and Kim [9] consider a similar production-inventory system with two machines connected by a finite buffer between them, where the objective is to determine the optimal buffer level that triggers PM. Das and Sarkar [10] consider a joint (s, S) inventory and failure-prone production system, and study PM policy based on the inventory level and the number of products made since last maintenance.

However, there are very few papers on PM scheduling under the specific context of semiconductor manufacturing. Recently, Mosley et al. [11] study maintenance dispatching and staffing policies for a group of fabs sharing maintenance resources. López and Wood [12] study the impact of configuration and maintenance policies on the performance of systems of cluster tools.

In semiconductor manufacturing, hierarchical planning and scheduling for maintenance activities is common practice. On the higher level, the PM frequency, or interval between two consecutive PM windows, is planned first, and on the lower level, each PM can be scheduled

within a window. For example, a calendar window for a PM task can be given as "14 days $\pm$ 1 day". Hence, 13-14-15 days since last PM would be, respectively, the warning-due-late dates. Although this is common in semiconductor manufacturing and in many other applications as well, this type of hierarchical PM scheduling structure has not been addressed formally until very recently.

The recent work by van Dijkhuizen and van Harten [13] closely resembles our proposed hierarchical framework. They study a two-stage maintenance policy, where the first stage is related to the higher level, with the objective to determine a time window, and the second stage is related to the lower level to determine the actual start time of a PM within the time interval. However, the problem setting is just for a single PM on one tool. Besides, they assume the time for a PM is negligible.

The value of consolidation of different PM tasks, e.g., for cluster tools, is commonly recognized in semiconductor manufacturing; yet it has not been addressed in a rigorous way in the literature. One study of the problem of grouping maintenance activities, which doesn't consider production costs, is conducted by Wildeman et al. [14], in a generic problem setting. They consider a multi-component system where preventive maintenance activities can be carried out on each component with a system-dependent cost, which is same for all activities, and a component-dependent cost. It is desirable to group maintenance activities since execution of a group of activities requires only one setup. A rolling horizon dynamic policy for grouping PMs is developed.

III. Hierarchical PM Planning and Scheduling Framework

In this section, after a discussion of the problem background, we present our proposed two-level hierarchical modeling framework for PM planning and scheduling in semiconductor manufacturing.

A. Background

Our proposed hierarchical framework was motivated by discussions with operation analysts and tool group managers in semiconductor manufacturing fabs. Although our proposed solutions are applicable to all tool groups in a fab, those groups with highly complex and interdependent PM tasks, and high utilization rates would clearly be most positively impacted. Commonly, groups with cluster tools fall in the latter category, and we focus on these to illustrate our

subsequent presentation. Cluster tools are highly integrated machines that can perform a sequence of semiconductor manufacturing processes. A general configuration of a cluster tool includes several processing chambers, load/unload locks, and transfer robots.

The scheduling of preventive maintenance for cluster tools is very complicated. To begin with, there are various PM activities on each component of the cluster tool. Roughly, they can be categorized into two types of PMs: calendar-based and operation-based. A calendar-based PM must be performed at some interval of calendar time, e.g., every 7, 14, 30, 90, 180 or 360 days. For an operation-based PM, the interval between two consecutive tasks is determined by the tool's operation history, which can be characterized by either wafer-count or cumulative operating time since the last PM. For example, for each processing chamber, a kit change is supposed to be undertaken at every specified number of wafers produced since the last PM. The vast majority of PM policies follow a "generalized age replacement" structure, in which a PM is scheduled for a time after a tool's "age" exceeds some threshold, but there is flexibility on the actual start time within some associated interval. Here, "age" means calendar-time or operation history, according to the type of PM. In semiconductor manufacturing practice, this is often called a "PM window" policy, where such a window is associated with each PM task. Even if a PM window is operation-based, e.g., "2000 wafers $\pm$ 10%" since last PM, tasks must be scheduled on a calendar basis, e.g., work shift and day. Furthermore, if an optimization model were to track wafer count, this would lead to a very high level of computational complexity, and scheduling decisions of the form "schedule PM task A at 1,860 wafer count", which would need to be converted to an equivalent calendar date. For these reasons, our models and algorithms operate on a calendar base, and PM window specifications are assumed to be given on this base. Converting operation-based data to equivalent calendar dates is commonly handled in ad hoc ways in practice. Efficient methods and algorithms have also been developed recently [15].

Another complexity in PM scheduling is due to the fact that there are several key factors affecting the decision-making process. First of all, careful coordination of PM tasks in different chambers is required in order to improve the entire tool's throughput. Usually, it is advantageous to consolidate PM tasks when another PM is planned on the near horizon, or tools are shut down due to unexpected events. Second, WIP has to be considered in the PM schedule. For example, it would be ideal to do PM in a period when WIP is low, and not to do PM in a period of high WIP or when many lots of wafers are scheduled to arrive. Finally, resource constraints such as

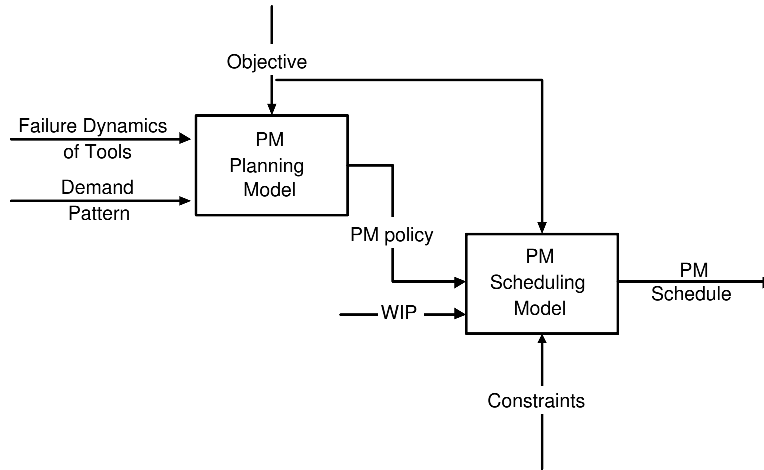

Fig. 1. Two-level hierarchical modeling framework for PM planing and scheduling

the headcount of maintenance technicians for the entire tool group of interest have to be taken into consideration, since manpower is usually the most critical constraint in PM scheduling.

In view of the complexity of PM scheduling in semiconductor manufacturing systems, we propose a two-level hierarchical modeling structure that can be applied to obtain "optimal" PM schedules from computationally tractable models. A similar idea is mentioned in [16], but without further development. Our hierarchical modeling framework is illustrated in Fig. 1, and we explain below the components of this framework.

B. PM Planning Model

The purpose of the PM planning model is to derive optimal policies for individual PMs. One candidate for such a policy could be an optimized time-window policy. Apart from conventional PM models [3], [4], [5], which include only the system's "technical" state information, i.e., deterioration degree or age of machines, our proposed PM planning models consider explicitly the system's "operating" states and "technical" states simultaneously. In the context of semiconductor manufacturing, the system's "operating" information, e.g., demand pattern, is as important as the system's "technical" information.

The planning model takes tool (stochastic) failure processes and incoming (stochastic) demand processes together with appropriate system objective functions as model inputs, and is formulated

as a Markov Decision Process (MDP) [17]. MDP methodology is a natural choice to deal with this type of problems, due to the feature of underlying stochastic processes and sequential decision epoches.

Under the hierarchical framework, the information about interdependence of PM tasks and resource constraints will be ignored intentionally in the planning model, and be left to the lower scheduling model.

C. PM Scheduling Model

Given the optimal policy, the scheduling model will determine the best time to do a PM by considering other factors that have been ignored in the PM planning model. For example, in the context of cluster tools, the scheduling model will consider the interdependence of different PM tasks in terms of their joint impact on the entire tool's throughput, as well as the match up between the tool's availability and projected incoming WIP. The scheduling model obtains optimal PM schedules, under some objective function and resource constraints. The scheduling model is formulated as a mixed integer program and is described in detail in the next section.

IV. Mixed Integer Programming of PM Schedules

A mixed integer programming model for PM scheduling for a group of tools is presented in this section. Specific issues on solving the MIP model and its software implementation will then be discussed.

A. Problem Definition

We consider PM scheduling for a group of tools. As explained in Section III-A, we assume all PM tasks are calendar-based. For the sake of generality, we give our presentation below in terms of cluster tools, keeping in mind that non-clustered tools can be viewed as single chamber tools for purpose of our model. Moreover, tools with coupled operations, e.g., litho steppers and trackers, should be modeled as a single tool with two chambers (in series).

Now consider a group of M cluster tools. The indexing of PM tasks for tool i is from 1 to ρ_i, where ρ_i is the total number of PM tasks applicable to tool i. For each cluster tool, the joint impact of PM tasks on its relative throughput, defined with respect to a fully operational tool, is characterized through a so-called "configuration matrix". Table I illustrates such a matrix,

which defines availability of tool as a function of its chamber statuses. Each row represents a scenario of chambers along with the corresponding availability of the entire tool. For example, the first row represents the scenario when all chambers 1 to 5 are up, indicated with "1", and so its availability is by definition 100%. The second and third rows represent the scenarios when either Ch1 or Ch2 is shut down, indicated with "0", for PM, respectively, with relative availability of only 60%. However, the 4th row shows that relative availability is 0 when both Ch1 and Ch2 are down at the same time, regardless of the status of all other chambers (indicated with "X"). This suggests that each wafer likely has to go through either Ch1 or Ch2, so when both chambers are down, no wafers can be processed. This also indicates that it is unwise to consolidate PM tasks for Ch1 and Ch2. Similarly, the last row suggests that each wafer has to go through Ch3, and so when it is down, there is no throughput, and its availability is therefore 0.

Similar to the "configuration matrix", a table describing resource requirements will list the resources required and duration for each PM and any consolidated PMs.

Now, given a set of PM tasks that need to be scheduled on these tools in a scheduling horizon, with each PM task associated with a time window in which the PM has to be started, the problem is to determine the best time for doing each PM, with the objective of maximizing overall tool availability and minimizing WIP, under some resource or operation constraints.

We formulate below the problem as an MIP model.

B. *MIP formulation*

Let t denote a generic time period, or PM decision epoch, and T the planning horizon; hence $t = 1, \ldots, T$. For example, time could be divided in periods of one work shift or one day, and the planning period could be two weeks, i.e., $T = 42$ shifts (assuming 3 shifts per day) or $T = 14$ days, respectively. The notation used throughout the paper is summarized in the following.

1) Indices

   i: tool

   l: PM task

   t: time unit

   j: resource type

2) Decision variables

TABLE I

CONFIGURATION MATRIX FOR A CLUSTER TOOL (LEGEND: 0: DOWN; 1: UP; X: IRRELEVANT)

Ch1	**Ch2**	**Ch3**	**Ch4**	**Ch5**	**Availability**
1	1	1	1	1	100%
0	1	1	1	1	60%
1	0	1	1	1	60%
0	0	X	X	X	0%
1	1	1	0	1	80%
1	1	1	1	0	80%
X	X	X	0	0	0%
1	0	1	1	0	60%
1	0	1	0	1	60%
0	1	1	1	0	60%
0	1	1	0	1	60%
X	X	0	X	X	0%

$a_i^l(t)$: binary decision variables for PM task l on tool i in period t, (1: do PM; 0: do not do PM). Define $\underline{a}_i(t) = [a_i^1(t)\ a_i^2(t)\ \ldots\ a_i^{\rho_i}(t)]^T$, the decision vector for all PM tasks on tool i.

$V_i(t)$: availability of tool i in period t.

$I_i(t)$: workload level (total in buffer and in process) for tool i in period t.

3) Parameters

M: number of tools considered

T: number of time units in the planning horizon

N: number of resource types considered

ρ_i number of PM tasks on tool i

w_i^l, u_i^l: time window [min, max] associated with PM task l on tool i.

k_i: number of periods for the PM task with the longest duration on tool i.

$d_i(t)$: projected incoming WIP for tool i in period t.

b_i: profit coefficient for availability of tool i.

c_i^I: cost coefficient for inventory on tool i.

c_i^l: PM cost for performing PM task l on tool i.

L_i: WIP buffer size for tool i.

K_i: coefficient of wafer throughput for tool i's availability.

$f_i(\cdot)$: availability function for tool i; constructed from the "configuration matrix", e.g., Table I.

$r_i^j(\cdot)$: resource function calculating the requirement of resource type j for tool i, $j = 1, \ldots, N$; constructed from a resource requirement matrix.

$R^j(t)$: amount of resource type j available in period t, $j = 1, \ldots, N$.

Our model is then given as follows.

<u>**Model MIP1**</u>:
$$\max \sum_{t=1}^{T} \sum_{i=1}^{M} \left(b_i \cdot V_i(t) - c_i^I \cdot I_i(t) - \sum_{l=1}^{\rho_i} c_i^l \cdot a_i^l(t) \right) \tag{1}$$

subject to:

$$\sum_{t=w_i^l}^{u_i^l} a_i^l(t) = 1, \text{for those PM tasks that have to be finished in the time window } [w_i^l, u_i^l] \subseteq [1, T]. \tag{2}$$

$$V_i(t) = f_i\left(\underline{a}_i(t), \underline{a}_i(t-1), \ldots, \underline{a}_i(t-(k_i-1))\right), \text{ for } i = 1, \cdots, M; t = 1, \cdots, T; \underline{a}_i(t) = 0, \text{for } t \le 0. \tag{3}$$

$$R^j(t) \ge \sum_{i=1}^{M} r_i^j\left(\underline{a}_i(t), \underline{a}_i(t-1), \ldots, \underline{a}_i(t-(k_i-1))\right), \text{ for } t = 1, \cdots, T; j = 1, \cdots, N; \underline{a}_i(t) = 0, \text{for } t \le 0. \tag{4}$$

$$I_i(t+1) = \left(I_i(t) - K_i \cdot V_i(t) + d_i(t)\right)^+, \text{for } i = 1, \cdots, M; t = 1, \cdots, T-1. \tag{5}$$

$$I_i(t) \le L_i, \text{for } i = 1, \cdots, M; t = 1, \cdots, T. \tag{6}$$

On (5), the operation $(\cdot)^+$ is defined as $(x)^+ = \max(0, x)$. The objective is to maximize profits from tool availability, minus costs from inventory build-up and performing the PM tasks. Equation (2) states that the scheduled PM tasks have to be performed within their individual time windows. Equation (3) computes the availability for each tool for each time period. A particular sequence of $\underline{a}_i(t), \underline{a}_i(t-1), \ldots, \underline{a}_i(t-(k_i-1))$, determines a particular row of the

"configuration matrix" and the value of $f_i(\cdot)$ would be the corresponding value of availability for that row. Equation (4) states that for each type of resource the sum of resource requirement over all tools must be less than available resource in each period. Equation (5) describes the WIP dynamics for each tool, and implies that for each tool i, it would produce as many wafers as possible, using all availability at hand, if there is enough in-buffer WIP; otherwise it would produce wafers matching up with the in-buffer WIP. Equation (6) states that the WIP level of tool i should not exceed its buffer size at any time.

Model parameters such as $b_i, c_i^I, c_i^l, L_i, R^j, K_i$, etc., are fab specific data, and can be obtained from fab operations. Specifically, the profit coefficient b_i could be determined as follows. First, for each tool i, we compute the wafer throughput per time unit (shift or day) assuming the tool is 100% available. Second, multiplying this by the average added value per wafer due to the operations on tool i, we obtain an estimated b_i. However, the cost coefficient c_i^I is relatively more difficult to compute, but could be determined by tool owners on the basis of how much impact a high WIP level would have on the performance of the fab. If the WIP level is not considered critical when below some threshold, then c_i^I can be set to zero.

In the above formulation, without loss of generality, we have assumed that during the scheduling horizon, each PM is performed at most one time on each tool, as reflected in Equation (2). This assumption does not affect PM tasks of the same type performed on different chambers, because they should have been indexed differently due to their association with different chambers. In the case when the same type of PM needs to be scheduled more than once for the same tool during the horizon, different indices should have been assigned to them, so that they will be treated as different PMs.

Equations (3) and (4) contain the respective availability and resource functions f_i and r_i^j. Albeit nonlinear in general, these can be easily implemented as look-up tables for computational purposes. Moreover, they can be also transformed into an equivalent set of linear equations, exploiting the fact that all arguments are binary; see the Appendix for details.

Note that the constraint (5) is non-linear, due to the operator $(\cdot)^+$. However, since it is piecewise linear, we define the following related problem:

Model $\mathbf{MIP1'}$: Same as $\mathbf{MIP1}$, but with (5) replaced by the following two linear constraints:

$$I_i(t+1) \geq I_i(t) - K_i \cdot V_i(t) + d_i(t), \text{for } i = 1, \cdots, M; t = 1, \cdots, T-1. \quad (7)$$

$$I_i(t) \geq 0, \text{for } i = 1, \cdots, M; t = 1, \cdots, T. \quad (8)$$

Proposition 1: $\mathbf{MIP1'}$ is equivalent to $\mathbf{MIP1}$.

Proof: See the Appendix. ■

It is worth noting here that in the objective function, minimizing the cost of a tool's WIP level implies maximizing its wafer throughput, due to the following relationship:

$$I_i(t+1) = I_i(t) - X_i(t) + d_i(t), \quad (9)$$

where $X_i(t)$ is the wafer throughput of tool i in the time period t, and is given by

$$X_i(t) = \min\{K_i \cdot V_i(t),\ I_i(t) + d_i(t)\}. \quad (10)$$

It can be easily seen that (5) is just a compact expression of (9) and (10).

There is a slight difference, though, between maximizing tool availability and maximizing wafer throughput. The former is relevant only to the tool technical state, i.e., keeping the tool operational as long as possible, whereas the latter is not only relevant to the tool technical state, but also strongly affected by projected incoming WIP. However, for bottleneck tools, maximizing availability is equal to maximizing wafer throughput, and vice versa, because there is always enough in-buffer WIP to be processed.

On the other hand, in situations where one specifically wants to maximize wafer throughput, the MIP formulation would become a little different, as defined by the following problem:

$$\underline{\text{Model }\mathbf{MIP2}\text{:}} \quad \max \sum_{t=1}^{T} \sum_{i=1}^{M} \left(b_i' \cdot X_i(t) - \sum_{l=1}^{\rho_i} c_i^l \cdot a_i^l(t) \right) \tag{11}$$

subject to:

$$X_i(t) \leq K_i \cdot f_i\left(\underline{a}_i(t), \underline{a}_i(t-1), \ldots, \underline{a}_i(t-(k_i-1))\right),$$
$$\text{for } i = 1, \cdots, M; t = 1, \cdots, T; \underline{a}_i(t) = 0, \text{for } t \leq 0. \tag{12}$$

$$I_i(t+1) = I_i(t) - X_i(t) + d_i(t), \text{for } i = 1, \cdots, M; t = 1, \cdots, T-1. \tag{13}$$

$$I_i(t) \geq 0, \text{for } i = 1, \cdots, M; t = 1, \cdots, T. \tag{14}$$

and constraints (2), (4) and (6), where b_i' in the objective function is the profit coefficient for wafer throughput of tool i.

In general, $\mathbf{MIP1}$ and $\mathbf{MIP2}$ are not equivalent.

C. *Implementation Issues*

In the following, we focus on models $\mathbf{MIP1}$ and $\mathbf{MIP1'}$, the latter of which is more amenable for practical implementation purposes. A model implemented in practice will usually have a simpler structure than the general formulation above. For example, it is not uncommon to consider a group of homogeneous (identical) cluster tools. In that case, all tools have the same physical structures and PM tasks. Hence, their availability functions are the same, as well as the resource functions. Thus $f_i(\cdot)$ and $r_i^j(\cdot)$ will reduce to $f(\cdot)$ and $r^j(\cdot)$, respectively. In addition, if manpower, i.e., the number of available maintenance technicians, is the only resource constraint of interest, then the resource vector $\underline{R}$ becomes a scalar.

In order to deal with the non-linearity of functions $f_i(\cdot)$ and $r_i^j(\cdot)$, we introduce a new set of decision variables – PM task vectors. A PM task vector contains a set of PM tasks, which could be consolidated and performed on a tool. At each time there is only one PM task vector active on each tool. Each task vector corresponds to one scenario of PM consolidation; see the Appendix for more details about the definition of PM task vector and the model transformation. The set of task vectors can be generated dynamically according to which PM tasks have to be scheduled in a given scheduling scenario. For a predefined planning horizon, only those PM tasks whose time-windows fall into the horizon will be taken into consideration.

Based on the MIP model described above, an optimal preventive maintenance scheduling

system has been implemented within a real fab setting with capability of optimizing PM tasks for any work center consisting of a group of tools. With different data interfaces, the system is integrated with other information systems in the fab in such a way that a specific instance of the MIP model can be generated automatically by extracting PM data from a tool maintenance database, and WIP information from a real-time dispatch system, or a fab simulation model. After an optimal schedule is found, a summary report is presented to users with information of projected availability and WIP of each tool in each period along the scheduling horizon. A comparison list with initial PM schedules and model-optimized schedules is also generated, and users can decide whether or not the model-optimized schedule will be put into effect. Further development of software tools used in the implementation of our models, and feasibility studies at several different fabs, are currently under way.

V. SIMULATION CASE STUDY

In order to evaluate the performance of the MIP scheduling model, we have conducted a preliminary simulation case study. The fab simulation model we adopted in the study is an existing (company) validated simulation model that is used in a real fab for production planning and scheduling purposes, and is based on Brooks Automation's AutoSched AP software [18] (ASAP for short).

We present only summary information for this preliminary case study, with sensitive fab-specific data removed. The work center has 11 cluster tools, i.e., $M = 11$. These tools are homogeneous in the sense that they can perform the same processing steps and have the same configuration, i.e., same processing chambers and robots. Coincidentally, there are 11 PM tasks of interest in the study on *each* tool, and they are indexed from 1 to 11. The "configuration matrix" of these tools is the one given in Table I. The longest duration of any PM task is 2 days, and the only resource we considered was the manpower (headcount) of available maintenance technicians. The resource function of PM task vectors, i.e., resource requirement for any joint or single PM tasks, is listed as a table with each row corresponding to a PM scenario, its duration and its resource requirement.

The time unit is one day, and the scheduling horizon is one week, i.e., $T = 7$ and $t = 1, 2, 3, 4, 5, 6, 7$. Those PM tasks that are to be scheduled in the horizon per tool are shown in Fig. 2 together with their time windows defined as a pair of (earliest_start_date, late_date). So,

for example, PM task 1 on tool 1 should be performed between Monday and Wednesday. PM task 5 on tool 1 is performed between Wednesday and Friday, while task 10 is between Friday and Sunday. For tool 8, the PM task 6 has to be performed on Monday as its time window is shrunk to a point in this specific case.

A specific MIP model instance was then generated from these PM tasks, and their individual time windows, along with all other relevant data, such as availability and resource requirement data sets, were fed into the scheduling system. The model instance was then solved. (For this simulation case study, the transformed MIP model has a total of 686 decision variables and 698 constraints. It takes about 2 minutes to find a solution using IBM OSL solver on an IBM RS6000 39H workstation.) The corresponding model outputs, i.e., optimal PM schedules for these tools along the scheduling horizon, are also shown in Fig. 2 as the asterisk points. One main feature that can be seen in the figure is that the optimal PM schedule tends to consolidate PM tasks, as on tools 1, 6, 9, and 11.

We simulated one week of fab operations with the two different PM schedules: the schedule that was actually performed in operations, which is referred to as the "reference" schedule, and the optimized, model-based schedule. Ten replications of such a simulation were made, averaging output results. Each PM task was modeled as a "PM order" in ASAP in both simulations. Two statistics, the average number of wafers completed on each tool and average number of WIP wafers on each tool, were computed. Fig. 3 shows the percentage throughput increase on each individual tool as well as the total number over all tools for the model-based schedule versus the reference schedule, i.e., $\mathrm{increase\%} = \frac{\mathrm{TP_{model}} - \mathrm{TP_{ref}}}{\mathrm{TP_{ref}}} \times 100\%$, where $\mathrm{TP_{model}}$ and $\mathrm{TP_{ref}}$ are the throughput under the model-based schedule and reference schedule, respectively. Fig. 4 shows the same for WIP levels.

The simulation results show that the model-based schedule outperforms the reference schedule in both performance metrics. Table II shows the mean values and 95% confidence intervals of throughput and WIP changes for each of the individual tools. The aggregated improvement of throughput over the entire group is only For throughput change, although the differences for 10 out of 11 tools are not statistically significant, the largest improvement for tool 1 of 13.9% is in fact statistically significant; for WIP change, the decrease of WIP level for tool 11 of -2.1% is also statistically significant.slightly more than 1.6%, but for bottleneck tools even a small percentage improvement can have a substantial economic impact. For example, assuming a fab's

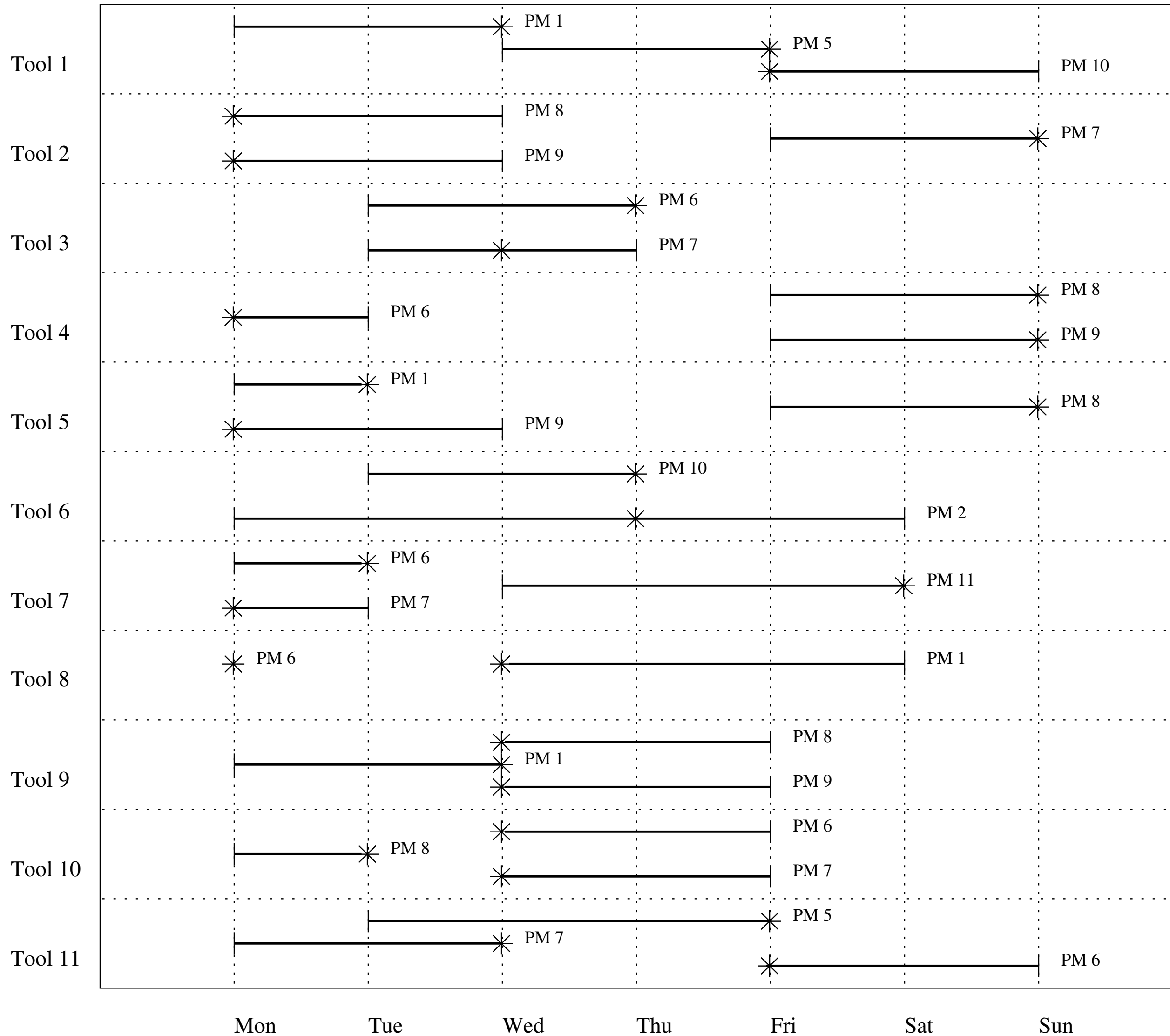

Fig. 2. PM tasks with associated time windows, where asterisk points are the optimal times computed by the model to perform PM tasks.

throughput is 5,000 wafers per week and the average price for a finished wafer is $15,000, then even a 1% improvement in throughput would result in a revenue increase of up to $750,000 per week, or about $39 million a year.

We foresee some enormous benefits will be resulted from an implementation of such model-based PM scheduling system. For example, the complicated scheduling process can be automated, and this will eventually increase the productivity of fab operations by saving human resources both in scheduling PMs and performing them. Closely related to this, the automated model-based

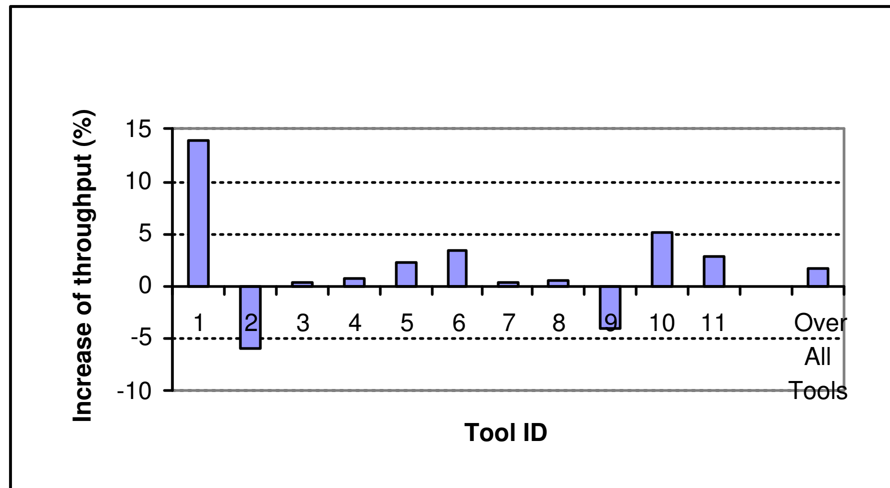

Fig. 3. Simulation result for throughput changes (in percentages) under the model-based schedule over those under the reference schedule.

TABLE II

MEAN VALUES AND 95% CONFIDENCE INTERVALS OF THROUGHPUT AND WIP CHANGES FOR INDIVIDUAL TOOLS.

	Throughput Change (%)		WIP Change (%)	
Tool ID	Mean Value	95% Confidence Interval	Mean Value	95% Confidence Interval
1	13.9%	[2.2%, 25.6%]	-1.8%	[-7.8%, 4.2%]
2	-5.9%	[-17.7%, 6.0%]	2.8%	[-1.4%, 7.1%]
3	0.3%	[-10.5%, 11.1%]	0.5%	[-1.1%, 2.0%]
4	0.8%	[-8.9%, 10.4%]	0.2%	[-2.9%, 3.2%]
5	2.2%	[-6.5%, 11.0%]	-0.7%	[-2.6%, 1.1%]
6	3.6%	[-2.9%, 10.1%]	0.9%	[-0.6%, 2.3%]
7	0.3%	[-10.9%, 11.6%]	0.4%	[-1.7%, 2.4%]
8	0.6%	[-10.2%, 11.4%]	-0.4%	[-2.3%, 1.5%]
9	-3.9%	[-16.3%, 8.4%]	-3.4%	[-11.6%, 4.8%]
10	5.3%	[-10.4%, 21.0%]	0.1%	[-2.5%, 2.7%]
11	2.8%	[-4.6%, 10.2%]	-2.1%	[-4.1%, -0.2%]

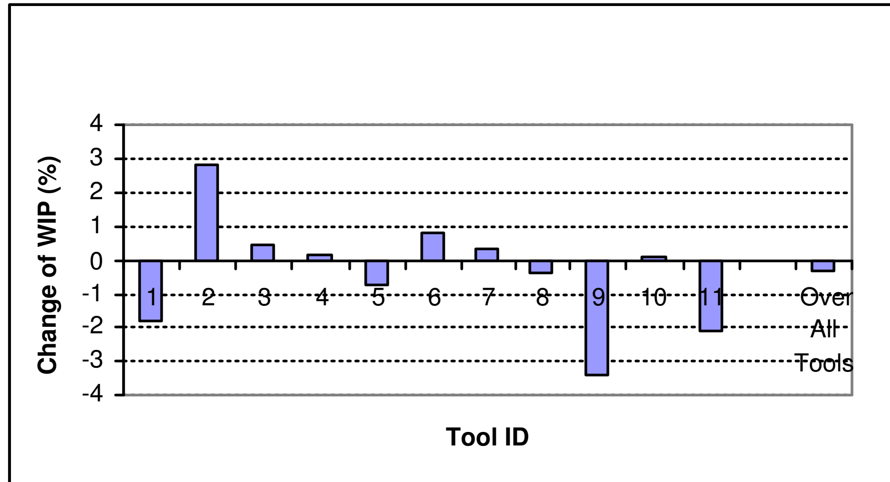

Fig. 4. Simulation result for WIP changes (in percentages) under the model-based schedule over those under the reference schedule.

PM scheduling should eliminate incidences of poor PM schedules due to occasional oversights in human judgment.

On the other hand, we surmise that the close performance between the model-based performance and the reference schedule can probably be attributed to the following factors. First, fab engineers should be given credits for doing a good job in PM scheduling on the basis of their rich experience in considering critical factors such as PM consolidation, and so they come up with a near-optimal reference schedule, especially for the most critical tools. Second, the benefits of model-based schedule might not have been revealed fully through the ASAP simulation, because of the simplified modeling structure of cluster tools as well as the fab model in ASAP. For instance, the relation between entire tool availability and chamber statuses can not be modeled in ASAP as precisely as in a "configuration matrix".

VI. Conclusions and Future work

In this paper, we studied the problem of PM planning and scheduling in semiconductor manufacturing. Due to the high level of problem complexity, we suggest a decomposition approach, i.e., a two-level hierarchical modeling framework. At the higher level is a model for long-term PM *planning*, which captures both the stochastic failure process of machines and

demand pattern of the system. At the lower level is a model for short-term PM *scheduling*, which conforms to PM policies from the planning model and obtains optimal PM schedule.

The paper focused on the lower level scheduling model, in the context of scheduling PM tasks over a group of cluster tools. We develop a mixed integer programming model, taking into account interdependence among PM tasks, resource constraints, and projected WIP data. By introducing new decision variables, non-linear functions appearing in the general MIP model can be transformed into linear functions, resulting in a model easily solvable using any commercial LP/IP software package.

Our study shows that the approach is feasible and promising. An implementation of such a model-based PM scheduling system could bring to a manufacturer several benefits. By providing an optimized schedule, the system has the potential to increase equipment availability and thus to generate more profits from fab operations, while eliminating human errors. However, effective implementation requires a certain information systems infrastructure be in place, e.g., a tool management system providing equipment status along with the raw schedule of PM tasks, and a wafer dispatching system that can estimate projected WIP levels.

Regarding the model scalability, a model involving a group of a dozen tools and a set of a dozen PM tasks on each tool seems fairly representative, and is easily in the solvable space of commercial solvers. However, the scaling issue might become an issue if a finer granularity of planning time unit is needed. For example, if the time unit is one hour, the model instance will easily exceed ten thousand decision variables. We suggest a unit of a day or an eight-hour shift, which is comparable with the frequency of most PM decisions.

Further development of software tools used in the implementation of our models, and feasibility studies at several different fabs, are currently under way. Regarding future work in the lower level scheduling model, there are several directions in which the model can be extended. One direction is to incorporate statistical process control (SPC) data into the model. The idea is that the PM schedule would be able to respond to possible “out of control” events, by triggering a “pull” or “push” of the planned time window of the corresponding PM. Another direction is to consider PM policies without time windows, with a penalty imposed if its starting time differs from a planned time. It would be easy to extend our developed MIP model to this case simply by removing time window constraints and adding a penalty function into the model objective. There is also some questions about the possible increase of variance introduced by consolidating

PM tasks. Such questions remains to be answered and are subjects of the future research.

ACKNOWLEDGEMENTS

The authors thank Javad Ahmadi, Matilda O'Connor, Mike Hillis, and Nipa Patel for their invaluable support. We thank Jason Crabtree and Jose Ramirez for their comments and suggestions on earlier versions of this work. We are also thankful for the suggestions from our industry liaisons in the FORCe program, the Associate Editor and anonymous referees.

VII. APPENDIX

A. Proof of Proposition 1.

Let $\mathcal{D}(\mathbf{P1})$ and $\mathcal{D}(\mathbf{P1}')$ be all feasible solutions of $\mathbf{P1}$ and $\mathbf{P1}'$ respectively. We first show $\mathcal{D}(\mathbf{P1}') \supseteq \mathcal{D}(\mathbf{P1})$. Note that the constraints (7) and (8) of $\mathbf{P1}'$ can be written combinatorially as

$$I_i(t+1) \geq (I_i(t) - K_i \cdot V_i(t) + d_i(t))^+, \text{for } i = 1, \cdots, M; t = 1, \cdots, T-1. \tag{15}$$

Indeed, it is a relaxed constraint (5) of $\mathbf{P1}$. So $\mathcal{D}(\mathbf{P1}') \supseteq \mathcal{D}(\mathbf{P1})$.

It can be easily verified that any optimal solution to $\mathbf{P1}'$ will achieve $I_i(t+1) = (I_i(t) - K_i \cdot V_i(t) + d_i(t))^+$, which implies this is also an optimal solution to $\mathbf{P1}$. To see this, assume there is an optimal solution such that $I^*_{i_1}(t_1+1) > (I_{i_1}(t_1) - K_{i_1} \cdot V_{i_1}(t_1) + d_{i_1}(t_1))^+$. Obviously, if we choose $I'_{i_1}(t_1+1) = (I_{i_1}(t_1) - K_{i_1} \cdot V_{i_1}(t_1) + d_{i_1}(t_1))^+ < I^*_{i_1}(t_1+1)$, it will achieve a larger objective value. Contradicted.

Thus $\mathbf{P1}$ and $\mathbf{P1}'$ are equivalent. ■

B. Solving the MIP model

There are two technical problems that must be addressed in order to solve the MIP model. To begin with, there may be PM tasks with a duration exceeding a single period. As seen in availability function f_i and resource function r_i^j, this results in the difficulty that chambers statuses (thus, the tool's state) will depend not only on PM tasks initiated in current time period t, but also on those unfinished PM tasks that were initiated in $t-1$, $t-2, \ldots,$ etc. One method to work around this problem is to introduce some "artificial" PM tasks as follows.

Assume PM l lasts for 3 periods. We introduce two "artificial" PMs l' and l'' such that l' must be performed in the next period following l, and l'' is following l', and we now treat l as a PM

task with a duration of only one period instead of three. This relationship can be formulated as "precedence" constraints as follows:

$$a_i^{l'}(t+1) = a_i^l(t), \tag{16}$$

$$a_i^{l''}(t+1) = a_i^{l'}(t). \tag{17}$$

Thus any PM task with a duration exceeding one period can be transformed into a sequence of PMs of one-period duration. However, please be aware that the introduction of "artificial" PMs could result in a significant increase in the number of decision variables, especially when many PM tasks last for multiple periods. In the following analysis, we will assume without loss of generality that no PMs have a duration exceeding one period.

The second difficulty is that the availability function f_i and resource function $\underline{r}_i$ are non-linear functions of chamber status. To deal with the non-linearity, the main idea is to transform these non-linear functions into linear form by changing decision variables. Observe that availability and resource functions can be expressed in "look-up" table form, as the "configuration matrix". Explicitly, if we denote the state of tool i (i.e., all chambers statuses, up or down) by s_i, then the availability function will become $f_i(s_i)$, and we denote its value by $f_i^{s_i}$. Now the function can be expressed as a data set $\{f_i^{s_i}\}$.

The decision variable in the MIP model is $a_i^l(t)$, i.e., to determine whether PM task l is conducted on tool i in period t, for every feasible l. This is equivalent to determining a set of PM tasks (task vector) conducted on tool i for every period t. Because there is a finite number of PM tasks, it is easy to obtain all combinations, i.e., vectors of these tasks. For example, if there are n tasks on tool i, then there are $2^n - 1$ task vectors, which include all possible combinations of these n tasks. We denote the task vector by v, and for the sake of simplicity, we assume every vector v is associated with only one tool, i.e., it can be only applied to a specific tool. We denote by $\mathcal{V}(i)$, the set of all feasible task vectors for tool i. The information of element tasks included in a vector v is contained in data $e(l,v)$, where $e(l,v) = 1$ if v contains l, $e(l,v) = 0$ otherwise.

Now, define new binary decision variables $z(i,v,t)$ for $v \in \mathcal{V}(i)$, where $z(i,v,t) = 1$ if task vector v is performed on tool i in the period t, $z(i,v,t) = 0$ otherwise. Obviously, for $v \notin \mathcal{V}(i)$, $z(i,v,t) = 0$. It is also obvious that on each tool in any period, there is only one vector that

can be active. So, the following new constraints on $z(i,v,t)$ will be enforced:

$$\sum_{v} z(i,v,t) \leq 1, \text{for } i = 1, \ldots, M; t = 1, \ldots, T. \tag{18}$$

The original decision variable $a_i^l(t)$ can be expressed as follows:

$$a_i^l(t) = \sum_{v \in \mathcal{V}(i)} z(i,v,t) \cdot e(l,v), \text{for } i = 1, \ldots, M; l = 1, \ldots, \rho_i; t = 1, \ldots, T. \tag{19}$$

Since tool state s_i is completely dependent on task vector v, their relationship can be characterized by $\delta(v, s_i)$, where $\delta(v, s_i) = 1$ if v changes the tool state to s_i, otherwise $\delta(v, s_i) = 0$. The availability function now can be expressed as a linear function of the control variable $z(i,v,t)$ as follows:

$$V_i(t) = \sum_{v \in \mathcal{V}(i)} \sum_{s_i} f_i^{s_i} \cdot \delta(v, s_i) \cdot z(i,v,t), \text{for } i = 1, \ldots, M, t = 1, \ldots, T. \tag{20}$$

Similarly the resource requirement of the tool is dependent only on task vector, the corresponding resource function can be expressed as a data set $\{r_i^{j,v}\}$. Hence, equation (4) can be written as:

$$R^j(t) \geq \sum_{i} \sum_{v \in \mathcal{V}(i)} r_i^{j,v} \cdot z(i,v,t), \text{for } j = 1, \ldots, N; t = 1, \ldots, T. \tag{21}$$

Thus, we are able to transform non-linear functions into linear functions of the new decision variables, and the transformed MIP model can be solved by a commercial IP/LP package. (In our implementation, we employ IBM's EasyModeler and OSL package.) Equations (18) and (19) are the new constraints added in the transformed MIP model. The number of new constraints due to equations (18) and (19) is $M \cdot T + \sum_{i=1}^{M} \rho_i \cdot T$.

The drawback of introducing the new task vectors is that it will have a set of decision variables with much larger size. There are many ways generating the set of task vectors. The basic requirement is that the set of v must cover all possibilities of PM consolidations. One of the easiest ways, which is the method used in our current implementation, is to list all possible combinations of PM tasks for each tool, and then put these combinations together and give them different index numbers one by one. This method ensures all possibilities of PM consolidation would be covered by the set, and its size is $\sum_{i=1}^{M} 2^{\rho_i} - M$. The actual number of new decision variables due to $z(i,v,t)$ is $\left(\sum_{i=1}^{M} 2^{\rho_i} - M\right) \cdot T$.

REFERENCES

[1] S. Kumar and P. R. Kumar, "Queueing network modeling in the design and analysis of semiconductor wafer fabs," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 548–561, Oct. 2001.

[2] R. Uzsoy, C. Y. Lee, and L. A. Martin-Vega, "A review of production planning and scheduling models in the semiconductor industry, part I: system characteristics, performance evaluation and production planning," *IIE Transactions*, vol. 24, pp. 47–60, 1992.

[3] W. P. Pierskalla and J. A. Voelker, "A survey of maintenance models: the control and surveillance of deteriorating systems," *Naval Research Logistics Quarterly*, vol. 23, pp. 353–388, 1976.

[4] C. Valdez-Flores and R. M. Feldman, "A survey of preventive maintenance models for stochastically deteriorating single-unit systems," *Naval Research Logistics*, vol. 36, pp. 419–446, 1989.

[5] D. I. Cho and M. Parlar, "A survey of maintenance models for multi-unit systems," *European Journal of Operational Research*, vol. 51, pp. 1–23, 1991.

[6] H. Wang, "A survey of maintenance policies of deteriorating systems," *European Journal of Operational Research*, vol. 139, pp. 469–489, 2002.

[7] K. C. So, "Optimality of control limit policies in replacement models," *Naval Research Logistics*, vol. 39, pp. 685–697, 1992.

[8] F. V. der Duyn Schouten and S. G. Vanneste, "Maintenance optimization of a production system with buffer capacity," *European Journal of Operational Research*, vol. 82, pp. 323–338, 1995.

[9] R. D. Meller and D. S. Kim, "The impact of preventive maintenance on system cost and buffer size," *European Journal of Operational Research*, vol. 95, pp. 577–591, 1996.

[10] T. K. Das and S. Sarkar, "Optimal preventive maintenance in a production inventory system," *IIE Transactions on Quality and Reliability Engineering*, vol. 31(6), pp. 537–551, 1999.

[11] S. A. Mosley, T. Teyner, and R. M. Uzsoy, "Maitenance scheduling and staffing policies in a wafer fabrication facility," *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 2, pp. 316–323, May 1998.

[12] M. J. Lopez and S. C. Wood, "Systems of multiple cluster tools: Configurations, reliability, and performance," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, pp. 170–178, May 2003.

[13] G. van Dijkhuizen and A. van Harten, "Two stage generalized age maintenance of a queue-like production system," *European Journal of Operational Research*, vol. 108, pp. 363–378, 1998.

[14] R. Wildeman, R. Dekker, and A. Smit, "A dynamic policy for grouping maintenance activities," *European Journal of Operational Research 99*, pp. 530–551, 1997.

[15] J. A. Ramirez-Hernández and E. Fernández-Gaucherand, "An algorithm to convert wafer to calendar-based preventive maintenance schedules for semiconductor manufacturing systems," 2003, submitted for publication.

[16] P. A. Scarf, "On the application of mathematical models in maintenance," *European Journal of Operational Research*, vol. 99, pp. 493–506, 1997.

[17] M. L. Puterman, *Markov Decision Processes*. New York: John Wiley & Sons, Inc., 1994.

[18] Brooks Automation, AutoSched AP. [Online]. Available: http://www.brooks.com/pages/231_autosched_ap.cfm