Production Scheduling in a Semiconductor Wafer Fabrication Facility Producing Multiple Product Types With Distinct Due Dates

Yeong-Dae Kim, *Member, IEEE*, Jae-Gon Kim, Bum Choi, and Hyung-Un Kim

***Abstract*—This paper focuses on production scheduling in a semiconductor wafer fab producing multiple product types that have different due dates and different process flows. In the wafer fab, wafer lots are processed on serial and batch processing workstations, each of which consists of parallel identical machines. Machines in serial processing workstations process wafer lots one by one, while those in batch processing workstations process several wafer lots of the same recipe at the same time. What need to be done for production scheduling are lot release control (to determine when and which wafer lot to release into the wafer fab), lot scheduling (to determine processing sequences of lots waiting in front of serial processing workstations) and batch scheduling (to determine which lots to process simultaneously as a batch and when to process batches on batch processing workstations). For these three decision problems, we develop several rules which use information such as order sizes (numbers of lots in orders) and processing status of the wafer lots. To evaluate these new rules, we use a simulation model in which the three decision problems are considered simultaneously. Simulation results show that the new rules work better than existing rules in terms of total tardiness of the orders.**

***Index Terms*—Batch scheduling, lot release control, lot scheduling, semiconductor wafer fabrication, simulation, tardiness.**

I. Introduction

LIKE FIRMS IN any other industry, semiconductor manufacturing companies must meet customers' demands in terms of quality, quantity, and due dates to survive in competitive business environments. This is especially true in systems producing application-specific integrated circuit (ASIC) chips, which have low-volume and high-variety characteristics. Meeting due dates is more important in such low-volume and high-variety settings than in mass production settings. Since wafer fabrication is one of the most complex production processes involving hundreds of operations with reentrant flows, it is necessary to develop good scheduling and control policies in wafer fabrication facilities (fabs) for an effective and efficient operation of manufacturing systems for ASIC chips.

In ASIC fabs, multiple product types are produced simultaneously and orders with different due dates arrive dynamically. In this paper, an order (for wafers) is specified by a due date, a type of wafers and an order size, i.e., the number of identical wafer lots. A wafer lot is a transportation unit consisting of the same type of wafers. Processing of an order is considered to be completed when all lots in the order are completed. In most of current wafer fabs, there are two types of workstations, batch processing workstations and serial processing workstations [1], [2] and each workstation consists of single or parallel identical machines. While serial processing workstations process wafer lots one by one, each machine in batch processing workstations processes several wafer lots simultaneously. However, certain lots cannot be processed together on batch processing workstations. For example, it may be impossible to process lots with different recipes together in the same batch, since lots with different recipes may require different processing times [3].

In this paper, three problems are considered for production scheduling and control in wafer fabs producing multiple product types that have different due dates and different process flows. They are lot release control, lot scheduling in serial processing workstations and batch scheduling in batch processing workstations. For lot release control, it is necessary to select a wafer lot to be released into the fab and to determine at which time it is released, while it is necessary for lot scheduling to determine processing sequences of wafer lots waiting in front of serial processing workstations. For batch scheduling, it is necessary to group wafer lots waiting in front of batch processing workstations into batches and determine sequences and timing for processing these batches. It is known that performance of a wafer fab is strongly affected by these scheduling and control decisions [2].

Lot release control problems have been dealt with in a few papers. For example, Wein [2], Glassey and Resende [4] and Kim *et al.* [5] suggest lot release rules used to determine the time to release a new wafer lot into the fab, called the workload regulating rule, starvation avoidance rule and parametric workload regulating rule, respectively. In these rules, information on the workload at a bottleneck workstation is used to release of a new wafer lot. These rules can be effectively used for wafer fabs producing wafers of a single product type which do not have specific due dates. However, many semiconductor wafer fabs (especially ASIC fabs) produce multiple types of products that have different due dates and different process flows. In such

Manuscript received March 31, 2000; revised November 27, 2000. This paper was recommended for publication by Associate Editor X. Xie and N. Viswanadham upon evaluation of the reviewers' comments. This work was supported in part by the Korea Science and Engineering Foundation by Grant 98-0200-08-01-2.

Y.-D. Kim and B. Choi are with the Department of Industrial Engineering, Korea Advanced Institute of Science and Technology (KAIST), Yusong-gu, Daejon 305-701, Korea (e-mail: ydkim@convex.kaist.ac.kr; choice@bomun.kaist.ac.kr).

J.-G. Kim was with the Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, Yusong-gu, Daejon 305-701, Korea. He is now with Entrue Consulting, LG-EDS Systems, Inc., Yoido-dong, Seoul 150–721 Korea (e-mail: jaegkim@lgeds.lg.co.kr or jaegon@bomun.kaist.ac.kr).

H.-U. Kim is with Hyundai Electronics Ind. Co., Ltd. Ichon, Kyungki-do, 467-860, Korea (e-mail: hyungun@hei.co.kr).

Publisher Item Identifier S 1042-296X(01)09745-2.

fabs, it is necessary to select a wafer lot to be released into the fab as well as to determine the release time. Kim *et al.* [6] suggest dispatching rules for the selection of a new lot that can be used in those fabs and show that the suggested rules work better than other rules in terms of the total tardiness.

In most previous studies on lot scheduling problems in wafer fabs, dispatching rules have been used for sequencing. Wein [2] shows that the performance of sequencing rules depends on both the type of input rules and the number of bottleneck workstations in the fab. Since photolithography workstations are generally considered as bottlenecks, lot scheduling problems focused on these workstations in most studies. Graves *et al.* [7] develop a cyclic scheduling method in which each workstation performs operations assigned to it exactly once in a cycle and Lou and Kager [8] present a scheduling rule in which higher priorities are given to lots that have been processed on photolithography workstations more times. On the other hand, Steve *et al.* [10] devise scheduling policies for reducing the mean waiting time and the standard deviation of cycle time and Kim *et al.* [5] suggest a rule which is devised to balance the workload of each loop, i.e., a set of operations (workstations) between two consecutive photolithography operations. Johri [9] gives a two-stage method in which operations for the wafers are partitioned into groups with almost equal workloads and then simple rules are used for scheduling wafer lots for each group. Since due dates of orders are not considered but throughput and cycle time are used as performance measures in most studies except for that of Kim *et al.* [6], they may not be used for fabs producing multiple product types with different due dates.

Batch scheduling problems have been dealt with in a few studies as well. Neuts [11] suggests a rule called the minimum batch size rule, in which processing of a set of jobs, i.e., a batch, is started when the number of waiting jobs becomes greater than or equal to a predetermined number, called the minimum batch size. For semiconductor manufacturing systems, Glassey and Weng [12] give a method for scheduling jobs of a single job family on a single batch processing machine. This work is extended to scheduling problems in a multi-product batch processing systems in Fowler *et al.* [13] and Robinson *et al.* [14]. They devise strategies/rules using the information on the timing of future job arrivals [13] and the information on the current and expected future workload status of both upstream and downstream workstations [14]. Fowler *et al.* [15] later extend their strategies for single server case to a multiple server case. On the other hand, Uzsoy [3] suggests optimal solution algorithms for the problem of scheduling wafer lots of different process recipes on a single and parallel batch processing machines. Recently, Kim *et al.* [5] devise a rule in which scheduling decisions are made in such a way that the workloads of batch processing workstations and their downstream workstations are well balanced. As in the earlier studies on the other two problems, due dates of orders are not considered for this problem either in most of previous research. Although Sanjay and Uzsoy [16] present a dynamic programming formulation and heuristics for a problem of minimizing total tardiness on a batch processing machine, they focus on the static problems, in which jobs to be processed are given in advance.

As surveyed in the aforementioned, in most of previous studies on production scheduling in semiconductor wafer fabs, the three scheduling problems, i.e., the problems of lot release control, lot scheduling and batch scheduling, are considered separately. In addition, cases in which there are multiple product types with distinct due dates are not considered. Although due dates of orders are considered in Kim *et al.* [6], the batch scheduling problem is not considered in their research. In this paper, we simultaneously consider the three decision problems for production scheduling in wafer fabs and suggest new rules for the problems with the objective of minimizing mean tardiness of orders. Through a series of simulation experiments, we compare these new rules with existing rules and find the best rule combination that gives the best results.

II. Lot Release Control

What need to be determined in the lot release control problem are which wafer lot should be released into the wafer fab first and when it should be. Due to the complexity of the problem, optimal solution methods have not been sought for, but heuristic rules have been suggested in previous research. In this paper, a heuristic solution approach is employed as well. In the suggested approach, lot release control is done in such a way that a wafer lot is selected first and then the time to release the lot is determined.

For the selection of a lot to be released, two groups of dispatching rules are used in this paper. One is a set of existing rules that are reported to give good performance for lot release control (see Kim *et al.* [6] for a comparison of dispatching rules used for lot release control in a wafer fab) and the other is a set of rules that are newly developed in this paper. Priorities of waiting lots are determined using these dispatching rules and a lot with the highest priority is selected. Table I gives priority functions used in the dispatching rules. Note that a lot with the smallest priority function value has the highest priority.

The earliest due date (EDD) rule is one of the simplest rules used in practice for cases in which due dates are considered and the modified operation due date (MOD) rule, which is devised by Baker and Kanet [17], is known to work well in job-shop scheduling problems. In MOD rule, denoted MOD1-R here ("R" stands for "release"), the highest priority is given to a wafer lot with the minimum modified operation due date. The modified operation due date is the minimum of the operation due date and the earliest time the operation can be completed. The operation due date is given to each wafer lot (not to each order) for its first operation considering its remaining work and the due date of the order associated with the lot. In addition to the typical MOD rule, we test its variation (MOD2-R) that uses a different method to calculate the remaining work of each lot. In MOD1-R, the remaining work of lot i, P_i, is calculated as the sum of processing times for operations of lot i. In MOD2-R, on the other hand, we take account of the fact that there may be parallel identical machines in each workstation and therefore a certain operation for wafers in a lot can be processed by multiple machines. In this rule, the remaining work of lot i, P_i' is calculated as the sum of processing times for operations for the lot, each divided by the number of machines for the operation. For more details, see Kim [18], in which the MOD2 rule was suggested with other variations for job shop scheduling problems in which there are multiple

TABLE I
DISPATCHING RULES FOR LOT RELEASE

Symbol	Meaning
p_{ij}	processing time for operation j of lot i
m_{ij}	number of machines that can process operation j of lot i
P_i	total processing time required to complete all operations of wafer lot i $(P_i = \sum_j p_{ij})$
P_i'	$= \sum_j p_{ij}/m_{ij}$
d_i	due date of lot i, i.e., due date of the order associated with lot i
t	current time, when the decision for lot release is to be made
W_i	estimated total waiting time of lot i in the bottleneck workstation(s), which has the maximum per-machine workload among all workstations in the fab
n_i	size (number of lots) of the order in which lot i is included
r_i	number of lots already released into the fab among the lots in the order in which lot i is included
x^+	$\equiv \max(0, x)$
a, b, c	parameters used in dispatching rules

Rules		Priority of lot i
Group A	EDD	d_i
	MOD1-R	$\max\{d_i - b\cdot(P_i - p_{i1}),\ t + P_i\}$
	MOD2-R	$\max\{d_i - b\cdot(P_i' - p_{i1}),\ t + P_i'\}$
	MSEC-R	$\{(d_i - P_i - W_i - t)^+ + 1\}/(t + P_i + W_i)$
Group B	OS	$d_i - P_i - t - W_i - a\cdot n_i$
	MOS	$d_i - P_i - t - W_i - a\cdot(n_i - r_i)$
	WOS	$(d_i - P_i - t - W_i - a\cdot n_i)\cdot(1 + c\cdot r_i/n_i)$

identical jobs for each order and each operation of the jobs can be processed by multiple identical machines. The MOD rules include a parameter, b, in the priority function, which, usually with a value no less than 1, is used to consider waiting time of a lot in queues when estimating time required to complete a lot. On the other hand, the modified slack over estimated completion time rule, denoted MSEC-R here, which is slightly modified from MSEC2 suggested by Kim *et al.* [6], uses the ratio of the modified slack time (to be defined below) to an estimated completion time to compute priorities of waiting lots. MSEC-R differs from MSEC2 in methods to calculate the modified slack time and the estimated completion time. In MSEC2, these values are computed based on the processing times at bottleneck workstations.

Dispatching rules in group B are newly suggested in this paper. In these rules, modified slack time is used with certain variations. The modified slack time of lot i (S_i') is obtained by abstracting estimated waiting time of lot i, W_i, from slack time of lot i, $d_i - P_i - t$, i.e., $S_i' = d_i - P_i - t - W_i$. Here, t is the current time when the decision for lot release is made and W_i is (estimated) total waiting time of lot i at bottleneck workstation, which is computed as the product of the average work-in-process inventory (WIP) level of the bottleneck workstation, the average processing time of a lot at the workstation and the number of times the lot visits the workstation. The order slack (OS) rule uses the *order slack* of each lot, which is defined as slack time of the order in which the lot is included. It is obtained by subtracting an estimated completion time of the order from the due date of the order. That is, it is computed as $S_i' - a \cdot n_i$, where S_i' is the modified slack time of the lot, n_i is the order size of the order associated with the lot and a is a parameter used to take account of waiting times of lots in the order. In OS, a higher priority is given to lots in a larger-sized order if other conditions are the same. This rule is based on the concept that it is better to release lots of a large-sized order early to meet the due date of the order since lead time of an order (not a lot) increases as the size of the order increases.

The other two rules in group B are obtained by modifying OS slightly. The modified order slack (MOS) rule is different from OS in that the number of lots in an order that are not released yet is used instead of the total number of lots of the order, that is, $n_i - r_i$ is used instead of n_i. On the other hand, in the weighted order slack (WOS) rule, a weighted order slack is used to compute priorities of waiting lots. The weighted order slack of lot i is defined as the order slack of the lot multiplied by $1 + c\cdot r_i/n_i$, where c is a parameter for which the value is to be determined. Note that the order slack of lot i is larger if more lots in the order corresponding to lot i are released into the fab. Therefore, priorities of lots in an order decrease as the number of released lots in the order increases. MOS and WOS were developed under the presumption that it is better to let lots of an order with more lots that have not been released yet be released first, since orders with more lots still to be released are generally more urgent.

Once a wafer lot to be released into the fab is selected, the release time of the lot should be determined. Researchers have suggested heuristic rules such as the uniform release rule (UNIF), the constant WIP rule (CONWIP), the workload regulating rule (WR) and the starvation avoidance rule (SA). In UNIF, a new lot is released into the fab in a constant rate independent of the current system states [19], while a new lot is released whenever a lot is completed in CONWIP [20]. In WR, a new lot is released when the sum of remaining processing times of all lots at bottleneck workstation(s) falls below a given critical value. In SA, on the other hand, a new lot is released when virtual inventory level at the bottleneck workstation falls down to a predetermined value. The virtual inventory at a workstation denotes wafer lots being worked on or waiting in front of the workstation and those that are expected to arrive at the workstation within a certain period of time. It is reported that WR and SA work better than UNIF and CONWIP [2].

In this paper, a variation of WR, called the modified workload regulating rule (MWR), is devised to determine the time to release a wafer lot into the fab. MWR is similar to WR in that a lot selected with an input release rule is released into the fab when the sum of remaining processing times of all lots in the queue of the bottleneck workstation falls below a critical value α in both rules. In MWR, however, the selected lot is released into the fab regardless of the workload in the bottleneck workstation, if the slack per remaining work (SLACK/RW) of the selected lot (say lot i) is less than a predetermined value, β, i.e., if $(d_i - P_i - t)/P_i < \beta$. MWR is developed based on the idea that one should release urgent lots (lots with small SLACK/RW values) into the fab as early as possible in order to meet due dates of the orders although the current workloads of machines in the fab are high. Note that due dates of lots (or orders) are

TABLE II
DISPATCHING RULES FOR LOT SCHEDULING

Symbol	Meaning
$p_{i\hat{j}}$	processing time for operation $\hat{j}$ of lot i, where operation $\hat{j}$ is the operation to be performed at the current workstation
R_i	remaining work of lot i, i.e., $R_i = \sum_{j \in J_i} p_{ij}$, where J_i is the set of operations of lot i that are not performed yet
R_i'	$= \sum_{j \in J_i} p_{ij}/m_{ij}$
W_i^r	estimated total waiting time of lot i at the bottleneck workstations, including the waiting times to be incurred at future visits
n_i^r	number of lots which have more remaining work than lot i among lots in the order associated with lot i
l_i	index of the last lot, the lot with the most remaining work, or the lot with the least progress among lots in the order corresponding to lot i
h_i	difference in remaining works of lots i and l_i
a, b, k, δ	parameters used in dispatching rules

Rules		Priority of lot i
Group A	MOD1-S	$\max\{d_i - b\cdot(R_i - p_{i\hat{j}}),\ t + R_i\}$
	MOD2-S	$\max\{d_i - b\cdot(R_i' - p_{i\hat{j}}),\ t + R_i'\}$
	COVERT	$-\left[\{1 - \frac{(d_i - R_i - t)^+}{k\cdot b\cdot R_i}\}\Big/ p_{i\hat{j}}\right]^+$
	MSEC-S	$\frac{(d_i - R_i - W_i^r - t)^+ + 1}{t + R_i + W_i^r}$
Group B	ES/RW1	$\max\left\{(d_i - R_i - W_i^r - t)^+ \Big/ (R_i + W_i^r),\ \delta\cdot p_{i\hat{j}}\right\}$
	ES/RW2	$\max\left[\{d_i - R_{l_i} - W_{l_i}^r - t + a\cdot\frac{h_i}{P_i\cdot(n_i^r+1)}\}^+ \Big/ (R_i + W_i^r),\ \delta\cdot p_{i\hat{j}})\right]$
	ES/RW3	$\max\left\{(d_i - R_{l_i} - W_{l_i}^r - t + a\cdot\frac{n_i^r}{n_i})^+ \Big/ (R_i + W_i^r),\ \delta\cdot p_{i\hat{j}}\right\}$

not considered in the existing rules given above, but they are explicitly considered in MWR. However, MWR may not be always good for other performance measures, such as WIP level and cycle time. For instance, if there are too many *urgent* lots in the fab, the WIP level may increase excessively and the cycle time of lots will increase as well. Therefore, it is necessary to select a value for β carefully considering characteristics (operational objectives, current workload, etc.) of the fab.

III. LOT SCHEDULING

The lot scheduling problem considered in this section is the problem of determining processing sequences of lots waiting in front of serial processing workstations. Since serial processing workstations (especially, photolithography workstations) are generally considered as bottleneck workstations, lot scheduling in those stations may be very important. In this paper, we develop dispatching rules for the lot scheduling problem. These dispatching rules are applied when the workstation becomes available, i.e., at least one of the machines in the workstation becomes available. In this paper, two groups of dispatching rules are tested for lot scheduling. One is a set of existing rules that are known to perform well and the other is a set of rules that are developed in this paper. Rules tested in this research are given in Table II with priority functions.

MOD1-S, MOD2-S, and COVERT are rules which are known to work well for job-shop scheduling. (See Panwalkar and Iskander [21], Blackstone *et al.* [22] and Kim [18] for more details of these and other dispatching rules.) In these rules, b and k are parameters used to take waiting time of lots into account when estimating completion times of the lots. MSEC-S is the rule (MSEC) which worked best for lot scheduling in the test of Kim *et al.* [6], in which various dispatching rules were compared for lot scheduling.

Three dispatching rules in group B are developed using information on the order sizes and processing status of the wafer lots, in addition to due dates of the orders and processing times and waiting times of the lots. In these rules, priorities of the lots are determined by estimated slack time per remaining work (ES/RW). A lot with the smallest value of ES/RW is selected to be processed next unless there are lots with negative ES/RW values. If there are lots with negative ES/RW values, a lot with the shortest processing time is selected. Note that the shortest processing time rule tends to minimize the total tardiness of jobs when all jobs are already late. In these rules, δ is a parameter with a sufficiently small value. Here, it is set to 0.0001.

The three rules are devised by using three different methods to estimate slack times of the lots. In ES/RW1, slack time of lot i, s_i, is estimated as $s_i = d_i - (R_i + W_i^r + t)$, where t is the current

time, when the scheduling decision is to be made and W_i^r is (estimated) total waiting time of lot i during its future visits to the bottleneck workstation. Here, W_i^r is computed as the product of the average WIP level of the bottleneck workstation, the average processing time of the lot at the workstation and the number of times still left for the lot to visit the workstation before it is completed. To estimate the slack time in ES/RW2 and ES/RW3, we use information on processing status of other lots that are included in the same order, i.e., the order associated with the lot being considered. In these rules, slack time of a lot is set to be equal to estimated slack time of the *last lot* in the same order plus *additional slack*, to take account of the fact that the lot has more slack time than the last lot. As defined in Table II, the last lot denotes a lot with the most remaining work among lots in the same order. Since tardiness of an order is determined by the lot to be completed last, processing of a lot can be delayed without affecting tardiness of the order if the last lot has much more remaining work than the lot. However, if there are many lots between the current lot and the last lot, processing of the current lot should not be delayed too long in order not to delay start times of lots following this lot. In ES/RW2, the additional slack increases as the difference in the remaining work of the current lot and the last lot increases and as there are fewer lots between the two lots. ES/RW3 is different from ES/RW2 in that the additional slack increases as there are more lots between the current lot and the last lot regardless of the difference in the remaining work of the two lots.

IV. Batch Scheduling

Batch processing workstations, such as those for deposition/oxidation and cleaning, can process multiple lots (called a *batch*) together. At each batch processing workstation, wafer lots are classified into families according to process recipes of the lots (wafer lots in a family require the same recipe) and wafer lots that belong to the same family can be grouped into a batch. Note that a family may consist of wafer lots of different product types if they require the same recipe. Each machine in a batch processing workstation can process wafer lots in the same family simultaneously as a batch up to its capacity. This capacity is called the *maximum batch size* of the machine or workstation.

In this paper, three heuristic methods are suggested for batch scheduling. When one of the machines in a batch processing workstation becomes available, these methods are used to select a family, to select wafer lots in the family to form a batch to be processed next on the available machine and to determine the time to start processing the batch. In the suggested heuristics, a family (and wafer lots) is selected using information on slack times of the lots. Then, whether wafer lots of the selected family should be started now or not is determined by a certain rule using information on future arrival times of lots from upstream workstations and the number of waiting lots of the selected family. If it is decided that processing of the selected family should not be started at the current time but should be delayed, another family is selected in the same manner (in order not to let the machine idle unnecessarily). If none of the available families can be started at the current time, the machine becomes idle until processing of one of the families can be started.

If processing of the selected family can be started without delay, lots to be processed as a batch must be selected among wafer lots currently waiting in the queue. If the number of lots waiting in the queue is greater than the maximum batch size, lots with shorter slack times are selected first until the number of selected lots becomes equal to the maximum batch size. Otherwise, all waiting lots are processed as a batch. The overall procedure of the suggested approach is given below followed by detailed descriptions of the three methods suggested in this paper. The following procedure is applied to each workstation at each scheduling decision point, i.e., when (a machine of) the workstation becomes available or a new lot or lots arrive at the queue of a currently idle workstation.

Overall Procedure of Batch Scheduling

Step 1) (Selecting a family) Select a candidate family of lots to be processed immediately using a method defined in each heuristic. If there is no family to be considered as a candidate, stop. Wait until the next decision point.

Step 2) (Determining whether to start processing) Determine whether or not wafer lots of the selected family should be started right away using a method defined by each heuristic. If it is decided that processing of the selected family should be delayed, discard the family from the set of candidate families and go back to Step 1.

Step 3) (Forming a batch and starting the operation) If the number of lots waiting in the queue is greater than the maximum batch size, lots with shorter slack times are selected first until the number of selected lots becomes equal to the maximum batch size. Otherwise, all waiting lots are processed as a batch.

A. Modified Minimum Batch Size Rule (MMBS)

For the selection of a family to be processed next, the average slack time of lots of each family is computed and a family with the least average slack time is selected. If the number of waiting lots of the family in the queue is greater than or equal to a predetermined value (for the family), called the minimum batch size (MBS) of the family, the lots are grouped into a batch to be processed right away. Otherwise, processing for the selected family is deferred until the number of waiting lots becomes equal to the MBS. If there is no family to be processed at the present time, the machine must wait until a new lot (or lots) arrives at the queue for the machine, when a new scheduling decision is made. Here, different MBSs are used for different families since the maximum batch sizes are generally different for different families even at the same batch processing workstation. Note that MBSs are set to the same value for all families in all workstations in the well-known MBS rule. After a series of preliminary tests, the MBS for each family is set to be equal to 30 percent of its maximum batch size at each batch processing workstation in MMBS.

B. Modified Dynamic Batching Heuristic (MDBH)

This method is modified from the dynamic batching heuristic (DBH) suggested by Glassey and Weng [12] for the objective of minimizing the total waiting time. In DBH, only a single job family is considered for a single batch processing machine and the time to start processing a batch is determined using the in-

formation on future arrival times of lots in such a way that estimated waiting time is minimized. On the other hand, MDBH can deal with the batch scheduling problem with multiple job families and the time to start processing a batch is determined in such a way that total *weighted* waiting time is minimized.

In MDBH, a family to be processed next on an available machine is selected using the average slack time of lots of each family; a family with the least average slack time is selected. Note that due dates of lots are not considered in DBH. Once a family to be processed is selected, whether to start processing lots in the selected family is to be determined. For this decision, we compare two alternative points of time to start processing: the current time (when the scheduling decision is made) and the time point when a new wafer lot or wafer lots arrive at the workstation. For each alternative, we compute waiting times of the lots currently waiting in the queue and those that will arrive next and check whether the total weighted waiting time of the lots will be increased (or decreased) if processing of the available lots is started without delay rather than deferred until a new lot(s) arrives. Decision on the start time is made using the result of this comparison.

Assume that a scheduling decision is to be made at time t_0. We compare two alternatives: to start processing a batch immediately and to wait until a new lot or lots arrive at the queue (at time t_1). If the machine starts processing without delay, lots arriving at time t_1 must wait until one of the machines in the workstation becomes available, that is, they should wait until $\min(t_0 + \hat{p}_i, t_a)$, where $\hat{p}_i$ is the processing time for the lot and t_a is the time when another machine becomes available. Therefore, if starting of processing is deferred until time t_1, waiting times of lots currently waiting in the queue will be increased by $t_1 - t_0$ but waiting times of lots arriving at time t_1 will be decreased by $\max\{0, \min(t_0 + \hat{p}_i, t_a) - t_1\}$, compared to those of the other alternative. Although waiting times of lots arriving after t_1 are changed as well, such changes are not considered in the decision on the start/delay of processing in this method. This is because it is not easy to estimate consequences of the decision exactly. Note that actual starting times of the lots arriving after t_1 depend on arrival of other lots and the procedure for selecting lots to be processed next as well as the availability of other machines in the workstation. Fig. 1 shows an example for changes in the number of lots of a selected family in a queue in case $t_0 + \hat{p}_i < t_a$.

To choose one from the above two alternatives for starting processing a batch, the sums of weighted waiting times of lots available at the time points (t_0 and t_1) are computed with weights proportional to the reciprocals of slack times of the lots. If the sum of weighted waiting times resulting from the first alternative (to start processing a batch without delay) is smaller, it is started right away. Otherwise, processing of a batch of the selected family is deferred until the next decision point. If there is no family to be processed at the present time, the machine must wait until a new lot arrives at the queue for the machine, when a new scheduling decision is made.

C. Processing Urgency Classification Heuristic (PUCH)

In this heuristic, information on processing urgencies (to be defined later) of lots and the number of waiting lots in the queue is used to select families to be processed first, to group wafer lots into batches and to determine processing sequences of these batches. Batch sizes are determined as in MMBS. However, in PUCH, very urgent lots can be processed as a batch although the number of lots may be smaller than the MBS prespecified for its family. If there are two or more families which can be processed immediately, processing priorities of these families are determined according to processing urgencies of the families.

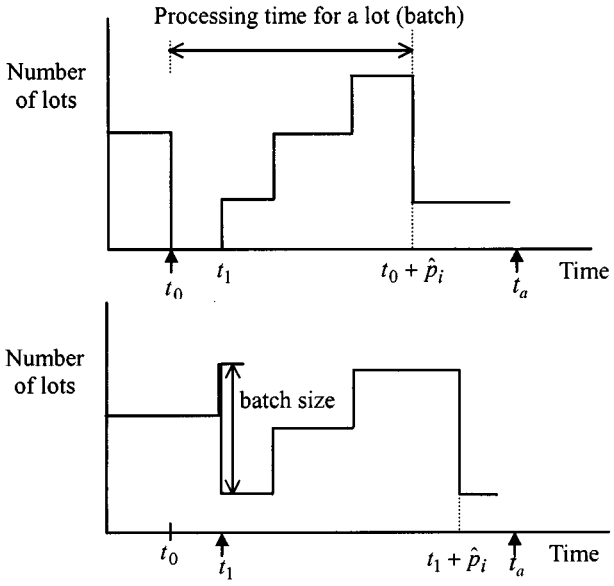

Fig. 1. Two alternative points of time to start a batch (the case in which $t_0 + \hat{p}_i < t_a$).

In this heuristic, families are classified into three classes according to the processing urgency. Families in class A are those that contain very urgent lots that have to be processed immediately or as soon as possible. Class-B families are those that do not contain very urgent lots but contain moderately urgent lots. The remaining families are classified as class C. Here, the urgency of a wafer lot is measured with slack per remaining work (SLACK/RW) of the lot, which is calculated as $(d_i - R_i - t)/R_i$ for lot i (see Tables I and II for the notation). If this value of a lot is less than or equal to γ_1, it is considered very urgent and if this value is less than γ_2 but greater than γ_1, it is considered moderately urgent. Values for γ_1 and γ_2 can be determined through simulation tests on several candidate values considering characteristics of the system or system states. In this paper, γ_1 and γ_2 are set to 1.5 and 3.0, respectively.

When a machine becomes available, if there are families of class A, a family which contains the most urgent lot, i.e., a lot with the least SLACK/RW, is selected and processed at once. If there is no class-A family, families in class B are considered. If there are class-B families for which the number of waiting lots is greater than or equal to the MBS, a family that contains the most urgent lot among them is selected and processed. Otherwise, we consider class-C families for which the numbers of waiting lots are greater than or equal to the MBS and select a family with the most waiting lots. If the number of waiting lots of the selected family is greater than the maximum batch size of the machine, lots with shorter slack times are selected first. Note that in this rule, an available machine is left idle if there is no family for

which the number of waiting lots is greater than or equal to the MBS, except for the case where there is a class-A family.

V. Simulation Experiments

In order to test the performance of the suggested rules, simulation experiments are carried out on a number of test problems randomly generated using a real fab data set prepared by Fegin, Fowler and Leachman, which is available at the website http://www.eas.asu.edu/~masmlab
. The configuration of the wafer fab and process plans (routes and processing time on each station) of the products used in the test are the same as those of the AISC fab corresponding to set 5 of the wafer fab data sets.

The following summarizes the configuration of the fab and how the test problems were generated.

1) There are 85 workstations (18 serial processing workstations and 67 batch processing workstations).
2) Each workstation consists of single or multiple identical machines.
3) There are 177 different products and these products are aggregated into 21 different types.
4) Wafer lots of the same product type have the same process plan. The size (number of wafers) of a lot is either 25 or 50.
5) The numbers of operations of the products range from 121 to 266 with an average of 182.
6) Each product requires 10∼15 layers and processing for each layer requires a loop of operations.
7) Time required for maintenance of the machines, break downs and yield losses due to scrap are not considered.

Table III shows product information of the fab data set including the total processing times, lot sizes and the number of operations required for each product type. The fab has a capacity of producing about 10 000 wafers per month and demand data in the test problems were generated in such a way that workload of the fab becomes approximately 95% on average. Note that if the load of the system is too low, there will be no difference in the performance of scheduling rules (all rules may result in zero tardiness).

We obtained 10 test problems by randomly generating 10 different sets of orders, each approximately amounts to the production quantity of 12 months. In each test problem, about 60 orders were generated for each month and interarrival times of the orders were generated from an exponential distribution. Each order consists of 5 to 7 identical wafer lots with the same due dates. The due date of order k is generated as $\hat{d}_k = \hat{a}_k + u \cdot \hat{P}_k$, where $\hat{a}_k$ is the time when order k arrives, $\hat{P}_k$ is the total processing times for the product type associated with order k and u is a parameter that defines tightness of due dates. For each order, u was set to a random number between 1 and 2, 2 and 4, or 4 and 6 with probabilities 0.2, 0.3 and 0.5, respectively.

Simulation tests were done on a personal computer with a Pentium processor (500 MHz) using FACTOR/AIM, a simulation software developed by Pritsker Corporation and additional user codes written in the C language. For each problem instance, a simulation run was made with the length of 12 months and results of the last 10 months were used for comparison. The results include the tardiness of orders that arrive after the beginning of the second month and that are due on or before the end of the twelfth month. For orders that are already tardy at the time a simulation run is terminated, tardiness of each order (say order k) is set to a lower bound, which is calculated as $t + \hat{R}_k - \hat{d}_k$, where t is the time when the simulation run is terminated and $\hat{R}_k$ is the remaining work of the last lot of order k.

TABLE III
Product Information

Product type	Total processing time (hours)	Lot size (wafers)	Number of operations
1	233	25	150
2	292	50	150
3	241	25	162
4	303	50	162
5	293	25	213
6	368	50	213
7	284	25	201
8	355	50	201
9	222	25	168
10	259	25	202
11	326	50	202
12	308	25	229
13	250	25	190
14	313	50	190
15	297	25	217
16	348	25	259
17	309	25	230
18	219	50	118
19	174	25	125
20	224	50	125
21	172	25	117

To evaluate performance of the rules tested in this paper, they were compared with benchmark solutions. These benchmark solutions were obtained using EDD or EDD-based rules for the three problems, which can be easily used in practice (or commonly used in many manufacturing systems), as follows.

1) Lot release control: EDD is used to select a lot to be released, while MWR is used to determine the time to release it.
2) Lot scheduling: EDD is used.
3) Batch scheduling: MBS+EDD, a combination of the MBS and EDD rules, is used. That is, if there are more than one family for which the number of waiting lots is greater than or equal to the MBS (the MBS is set to 2 for all families after tests on a few problems), a family that contains a lot with the earliest due date is selected. If the number of waiting lots of the selected family is greater than the maximum batch size of the family, the EDD rule is used to select lots to be processed next.

Solutions obtained from the scheduling methods (rules) were evaluated with the percentage reduction of the solutions from the benchmark solutions. The percentage reduction of rule r is computed as $100(S_B - S_r)/S_B$ for each problem, where S_r is the solution value (mean tardiness) of rule r and S_B is the solution value of the benchmark solution for the problem.

Since many rules are tested for each decision problem in this paper, comparison of all combinations of the rules may require

TABLE IV
SELECTED VALUES FOR THE PARAMETERS USED IN DISPATCHING RULES

Dispatching rule		Parameter value(s)
Lot release	MOD1-R	$b = 1$
	MOD2-R	$b = 2$
	OS	$a = 10$
	MOS	$a = 10$
	WOS	$a = 10, c = 0.5$
	MWR	$\alpha = 2800, \beta = 1.3$
Lot scheduling	MOD1-S	$b = 1$
	MOD2-S	$b = 2$
	COVERT	$b = 1, k = 1$
	ES/RW2	$a = 30$
	ES/RW3	$a = 20$

excessive computation time. Moreover, some of the rules include parameters for which the values should be carefully determined for maximum performance of the rules. To reduce the computational burden, experiments were done in the following three steps: determination of the most appropriate parameter values for each rule; selection of a subset of rules for each decision problem; and comparison of all combinations of the selected rules. In the first step, to determine parameter values for lot release rules, we used EDD and MBS+EDD for lot scheduling and batch scheduling, respectively. In addition, to determine parameter values for lot scheduling rules, we used EDD and MBS+EDD for lot release and batch scheduling, respectively. In the second step, to select of a subset of rules for lot release, we used MOD1-S and ES/RW1 for lot scheduling and MMBS and MDBH for batch scheduling. On the other hand, to select a subset of rules for lot scheduling, we used MOD1-R and MSEC-R for lot release and MDBH and MMBS for batch scheduling, while we used MOD1-R and MSEC-R for lot release and MOD1-S and ES/RW1 for lot scheduling to select a subset of rules for batch scheduling. In the first two steps, a smaller number of problems were tested with a shorter simulation run time (8 or 10 months) to save time. It was confirmed after the main tests, i.e., the third step of the experiment, that results obtained with such few test problems with shorter simulation time were not much different from those of the main tests.

Table IV shows parameter values selected for the rules for lot release control and lot scheduling, which were obtained as results of the first step of the computational experiments. Using the selected parameter values for the rules, another series of preliminary tests were done to select rules that give better performance than others for each decision problem. Results of the tests are shown in Table V, which gives the average percentage reduction for each rule. All rules except for MOD2-S gave significantly better solutions than the rules used for the benchmark solutions. For the main tests, we selected three best performing rules for lot release control (MSEC-R, MOD2 and OS) and three rules for lot scheduling (COVERT, ES/RW2 and ES/RW1) and two rules for batch scheduling (MDBH and PUCH). Although ES/RW3 does not seem to be significantly outperformed by the three selected rules for lot scheduling, it was excluded in the main tests to save computation time (it did not outperform and hence is not expected to outperform the three selected rules).

TABLE V
RESULTS OF PRELIMINARY TESTS

Rules		Percentage Reduction (%)
Lot release	MSEC-R	47.7
	MOD2-R	44.5
	OS	40.2
	WOS	32.3
	MOS	30.6
	MOD1-R	27.2
	EDD	26.7
Lot scheduling	COVERT	73.5
	ES/RW2	69.7
	ES/RW1	69.5
	ES/RW3	67.7
	MSEC-S	45.2
	MOD1-S	28.7
	MOD2-S	−5.02
Batch scheduling	PUCH	69.0
	MDBH	64.8
	MMBS	46.2

TABLE VI
PERFORMANCE OF RULE COMBINATIONS

Scheduling rules		Lot release rule		
		MOD2-R	MSEC-R	OS
COVERT	MDBH	66.0%† (12.2)	66.8% (9.8)	67.7% (10.5)
	PUCH	71.7 (8.9)	71.3 (6.5)	72.9 (8.3)
ES/RW1	MDBH	72.0 (6.0)	75.1 (5.9)	75.1 (7.1)
	PUCH	71.3 (7.0)	69.2 (7.1)	68.0 (7.0)
ES/RW2	MDBH	75.3 (8.2)	75.5 (7.6)	75.7 (9.3)
	PUCH	70.3 (9.2)	73.5 (11.2)	70.7 (10.7)

† Average and standard deviation (in parenthesis) of the percentage reduction from benchmark solutions

Results shown in Table V imply that the performance of the scheduling system is affected by lot scheduling rules and batch scheduling rules more than by lot release control rules tested here.

In the main simulation experiment, we test all 18 combinations of three rules (MOD2-R, MSEC-R, OS) for lot release, three rules (COVERT, ES/RW1, ES/RW2) for lot scheduling and two rules (MDBH, PUCH) for batch scheduling on the 10 test problems. Results are given in Table VI, which shows the average and standard deviation of percentage reduction of mean tardiness (from that of the benchmark solution) obtained from each rule combination. In addition, Table VII shows the overall performance of the rules for each decision problem. As can be seen from the tables, there does not seem to be much difference in the performance of the rules. Note that percentage reductions obtained by the rules given in Table VII are different from those in Table V, since rules included in the tests associated with the two tables are different. That is, the test associated with Table VII includes only good rules, while the (preliminary) test associated with Table V includes rules that are not as good as the rules included in the other test.

TABLE VII
THE OVERALL PERCENTAGE REDUCTION OBTAINED FROM THE RULES

Rules		Percentage Reduction (%)
Lot release control	MSEC-R	71.90 (8.49)†
	OS	71.70 (9.11)
	MOD2-R	71.12 (8.88)
Lot scheduling	ES/RW2	73.50 (9.32)
	ES/RW1	71.79 (6.96)
	COVERT	69.41 (9.50)
Batch scheduling	MDBH	72.14 (9.22)
	PUCH	71.00 (8.34)

† Average and standard deviation (in parenthesis) of the percentage reduction from benchmark solutions

TABLE VIII
ANALYSIS OF VARIANCE

Source of variance†	Degree of freedom	Sum of squared error	Mean squared error	F-value
LR	2	19.58	9.79	0.13
LS	2	509.58	254.79	3.45*
BS	1	58.77	58.77	0.80
LR × LS	4	41.06	10.26	0.14
LR × BS	2	39.59	19.79	0.27
LS × BS	2	897.70	448.85	6.08**
Error	166	12250.35	73.80	
Total	179	13816.62		

† LR, LS, and BS denote lot release rules, lot scheduling rules and batch scheduling rules, respectively.
* There is difference at the significance level of 0.05
** There is difference at the significance level of 0.01

To show the effects of the rules and identify difference in the performance, an analysis of variance (ANOVA) was done and the results are given in Table VIII. The results show that there is a significant difference in the performance of the lot scheduling rules and there is an interaction effect between lot scheduling rules and batch scheduling rules. However, input release rules have neither meaningful main effect nor interaction effects with rules for the other two decision problems. Possible reasons for this indifference are that input sequences obtained by the rules were not much different and the average time interval between lot releases is very small in this specific fab situation. However, a major or most influential cause for the indifference may be that the main test includes only a subset of good performing rules for each problem.

When interaction effect is large, corresponding main effects alone have little practical meaning. That is, knowledge of the interaction is more useful than knowledge of the main effects [23]. Therefore, it is more meaningful to identify a combination of rules that gives good performance. To compare rule combinations for lot scheduling and batch scheduling, paired-t tests are done. The rules for lot release control are not considered in the tests, since they do not seem to have significant effect on the system performance. Results of the paired-t tests given in Table IX show that there is no significant difference among the four best rule combinations, (ES/RW2, MDBH), (ES/RW1, MDBH), (COVERT, PUCH) and (ES/RW2, PUCH), where the first and the second terms in each parenthesis denote lot scheduling rule and batch scheduling rule, respectively. These four rule combinations significantly outperformed the other two combinations. It seems that MDBH works well with ES/RW1 and ES/RW2 while PUCH goes well with COVERT.

TABLE IX
RESULTS OF THE PAIRED-T TEST

Rule combinations	PR (%)†	c1	c2	c3	c4	c5
c1 (ES/RW2, MDBH)	75.5 (10.5)					
c2 (ES/RW1, MDBH)	74.1 (10.1)					
c3 (COVERT, PUCH)	72.0 (8.1)					
c4 (ES/RW2, PUCH)	71.5 (6.9)					
c5 (ES/RW1, PUCH)	69.5 (7.7)	**	**			
c6 (COVERT, MDBH)	66.8 (6.3)	**	**	*		

† Average and standard deviation (in parenthesis) of the percentage reduction from benchmark solutions
* There is difference in the performance at the significance level of 0.05
** There is difference in the performance at the significance level of 0.01

From the results given above, we can argue that the suggested rules outperform existing rules for the three decision problems in terms of mean tardiness of the orders. The outperformance may be due to the fact that the newly suggested rules use information such as order sizes and processing status of wafer lots in addition to due dates of orders and hence tardiness of orders resulting from a certain job sequence can be estimated more accurately. Although the system performance (tardiness of orders) seems to be more affected by lot scheduling rules and batch scheduling rules than input release rules in the fab considered here, this may be because the main tests include only a subset of good release rules that work almost equally well. In conclusion, ES/RW2 and MDBH are recommended for lot scheduling and batch scheduling, respectively, since this combination gives better results than other combinations (although the difference is not very significant in some cases). Moreover, these two rules work consistently better than others in each of the two decision problems for various combinations of rules for the other decision problems.

VI. CONCLUDING REMARKS

We considered three production scheduling problems in a semiconductor wafer fab producing multiple products with the objective of minimizing mean tardiness of orders for the products. For the three problems, i.e., the problems for lot release control, lot scheduling at serial processing workstations and batch scheduling at batch processing workstations, new rules were developed using the information on the wafer lots and the states of the fab. They were compared with existing rules for scheduling in wafer fabs and well-known dispatching rules for job-shop scheduling. Results of a series of simulation experiments showed that the newly developed rules outperformed existing rules devised for production scheduling in other fabs or for job-shop scheduling. Also, it is found that there is an interaction effect between rules for lot scheduling and batch scheduling and hence the three (or at least these two) problems should be considered simultaneously in the research on production scheduling in wafer fabs.

As in other research on heuristic scheduling methods for various manufacturing systems, there certainly are limitations in

this research. For example, only a few rules were developed and tested in this paper although there may be almost unlimited number of rules or methods that can be devised using various information of the wafer lots or the wafer fab. Another drawback of this paper is that the simulation study was done using only one real fab data set. In addition, machine breakdowns, maintenance and variable yield rates should be considered in the simulation model to make the model more realistic. Performance of the rules suggested in this research can be further improved if one can use more accurate information on future states of the fab such as WIP levels at downstream workstations after certain time periods. This requires development of a model or a method that can predict future states of the fab more accurately.

REFERENCES

[1] H. Gurnani, R. Anupindi, and R. Akella, "Control of batch processing systems in semiconductor wafer fabrication facilities," *IEEE Trans. Semicond. Manufact.*, vol. 5, pp. 319–328, 1992.

[2] L. M. Wein, "Scheduling semiconductor wafer fabrication," *IEEE Trans. Semicond. Manufact.*, vol. 1, pp. 115–130, 1988.

[3] R. Uzsoy, "Scheduling batch processing machines with incompatible job families," *Int. J. Prod. Res.*, vol. 33, pp. 2685–2708, 1995.

[4] C. R. Glassey and M. C. G. Resende, "Closed-loop job release control for VLSI circuit manufacturing," *IEEE Trans. Semicond. Manufact.*, vol. 1, pp. 36–46, 1988.

[5] Y.-D. Kim, D.-H. Lee, J.-U. Kim, and H.-K. Roh, "A simulation study on lot release control, mask scheduling and batch scheduling in semiconductor wafer fabrication facilities," *J. Manufact. Syst.*, vol. 17, pp. 107–117, 1998.

[6] Y.-D. Kim, J.-U. Kim, S.-K. Lim, and H.-B. Jun, "Due-date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility," *IEEE Trans. Semicond. Manufact.*, vol. 11, pp. 155–164, 1998.

[7] S. C. Graves, H. C. Meal, D. Stefek, and A. H. Zeghmi, "Scheduling of re-entrant flow shops," *J. Operat. Manage.*, vol. 3, pp. 197–207, 1983.

[8] S. X. C. Lou and P. W. Karger, "A robust production control policy for VLSI wafer fabrication," *IEEE Trans. Semicond. Manufact.*, vol. 2, pp. 159–164, 1989.

[9] P. K. Johri, "Practical issues in scheduling and dispatching in semiconductor wafer fabrication," *J. Manufact. Syst.*, vol. 12, pp. 474–485, 1998.

[10] C. H. L. Steve, D. Ramaswamy, and P. R. Kumar, "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Trans. Semicond. Manufact.*, vol. 7, pp. 374–388, 1994.

[11] M. F. Neuts, "A general class of bulk service queue with poisson input," *Ann. Math. Stat.*, vol. 38, pp. 759–770, 1967.

[12] C. R. Glassey and W. W. Weng, "Dynamic batching heuristic for simultaneous processing," *IEEE Trans. Semicond. Manufact.*, vol. 4, pp. 77–82, 1991.

[13] J. W. Fowler, G. L. Hogg, and D. T. Phillips, "Control of multiproduct bulk service diffusion/oxidation processes," *IIE Trans.*, vol. 24, pp. 84–96, 1992.

[14] J. K. Robinson, J. W. Fowler, and J. F. Bard, "The use of upstream and downstream information in semiconductor batch operations," *Int. J. Prod. Res.*, vol. 33, pp. 1849–1869, 1995.

[15] J. W. Fowler, G. L. Hogg, and D. T. Phillips, "Control of multiproduct bulk server diffusion/oxidation processes. Part 2: Multiple servers," *IIE Trans.*, vol. 32, pp. 167–176, 2000.

[16] V. M. Sanjay and R. Uzsoy, "Minimizing total tardiness on a batch processing machine with incompatible job families," *IIE Trans.*, vol. 30, pp. 165–178, 1998.

[17] K. R. Baker and J. J. Kanet, "Job shop scheduling with modified due dates," *J. Operat. Manage.*, vol. 4, pp. 11–22, 1983.

[18] Y.-D. Kim, "A comparison of dispatching rules for job shops with multiple identical jobs and alternative routeings," *Int. J. Prod. Res.*, vol. 28, pp. 953–962, 1990.

[19] C. R. Glassey and M. G. C. Resende, "A scheduling rule for job release in semiconductor fabrication," *Operat. Res. Lett.*, vol. 7, pp. 213–217, 1988.

[20] M. L. Spearman, D. L. Woodruff, and W. J. Hopp, "CONWIP: A pull alternative to kanban," *Int. J. Prod. Res.*, vol. 28, pp. 879–894, 1990.

[21] S. S. Panwalkar and W. Iskander, "A survey of scheduling rules," *Operat. Res.*, vol. 25, pp. 45–61, 1977.

[22] J. H. Blackstone Jr, D. T. Phillips, and D. L. Hogg, "A state-of-the-art survey of dispatching rules for manufacturing jobshop operations," *Int. J. Prod. Res.*, vol. 25, pp. 1143–1156, 1982.

[23] D. C. Montgomery, *Design and Analysis of Experiments*, 3rd ed. New York: Wiley, 1991.

Yeong-Dae Kim (M'95) received the B.S. degree from Seoul National University, Seoul, Korea, and the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, both in industrial engineering, and the Ph.D. degree in industrial and operations engineering from the University of Michigan.

He is a Professor with the Department of Industrial Engineering, KAIST.

Jae-Gon Kim received the B.S., M.S., and Ph.D. degrees in industrial engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, in 1994, 1996, and 2001, respectively.

Currently, he is with LG-EDS Systems, Seoul, Korea, working as a consultant. His research interests include manufacturing system design, VLSI layout design, logistics, and production planning and scheduling.

Bum-Choi received the B.S. and M.S. degrees in industrial engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejon.

He is a Ph.D. student with the Department of Industrial Engineering, KAIST. His research interests include manufacturing system modeling, production planning and scheduling, and logistics.

Hyung-Un Kim received the B.S. degree from Chonnam National University, Kwangju, Korea, and the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, both in industrial engineering.

He is currently a System Engineer with the Hyundai Electronics Ind. Co., Ltd., Ichon, Korea. His research interests include production planning and scheduling in semiconductor manufacturing systems and simulation.