Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

MULTI-PRODUCT LOT MERGING/SPLITTING ALGORITHMS FOR A SEMICONDUCTOR WAFER FABRICATION

June-Young Bang
Jae-Hun Kang
Bong-Kyun Kim
Yeong-Dae Kim

Department of Industrial Engineering
Korea Advanced Institute of Science and Technology
Yuseong-gu, Daejeon 305-701, KOREA

ABSTRACT

This paper focuses on a lot merging/splitting problem in a semiconductor wafer fabrication facility. In the fab, two or more lots can be merged into a single lot if routes and all the processing conditions of the lots are the same for a number of subsequent operations, and the merged lot is split into the original lots at the point where the routes or processing conditions become different. We suggest lot merging/splitting algorithms to reduce the total tardiness of orders and the cycle times of the lots. The suggested algorithms are evaluated through a series of simulation experiments and the result shows that the algorithms work better than a method used in a real fab.

1 INTRODUCTION

To survive in a competitive market environment in the semiconductor manufacturing industry, companies should operate their manufacturing facilities at the maximum effectiveness and efficiency. Innovation on production management is also needed for meeting customers' demands, in terms of quantity and due date, and getting shorter lead time so that the companies can maintain or increase their market share. Especially, optimizing the operations of a system producing non-memory type products, such as system LSI (large-scale integrated-circuits) and application-specific integrated circuit (ASIC) products, is regarded to be harder than those producing memory-type products due to low-volume and high-variety characteristics. In this paper, we consider a production scheduling and control problem in a semiconductor wafer fabrication facility (fab) producing multiple product types. We focus on a lot merging/splitting problem and present lot merging/splitting algorithms for the objective of minimizing total tardiness of orders and the cycle times of wafer lots.

In the fab considered in this study, two or more lots can be merged into a single lot if routes and all the processing conditions of the lots, such as temperature and gas types, are the same for a number of subsequent operations. This merged lot needs to be split into the original lots at the point where the routes or processing conditions become different. There is another condition for lot merging, the capacity constraint of a lot. That is, the number of wafers to be included in a merged lot cannot exceed the capacity of a lot. Usually, in semiconductor fabs, wafers are processed in a lot of 25 wafers or less, which means the capacity of a lot is 25. In the fab considered in this study, 30% to 50% of lots are composed of less than 25 wafers. Throughout the paper, we call lots containing 25 wafers *full lots*, and lots containing 24 or less wafers *partial lots*. Since partial lots contain smaller numbers of wafers, the processing time of a partial lot is shorter than that of a full lot on serial processing machines, such as metal implantation and photolithography, even for the same product type. However, the processing time of the two lots may be the same on batch-processing machines such as those for chemical etching and cleaning.

There have been a number of research articles on production scheduling and control problems in semiconductor manufacturing systems, such as problems of lot release control, lot scheduling in serial-processing workstations, and batch scheduling in batch-processing workstations. Various lot release rules have been developed in many studies including those of Wein (1988), Glassey and Resende (1988) and Kim et al. (1998a). In these rules, information on the workload at a bottleneck workstation is used for lot release. In most previous studies on lot scheduling problems in wafer fabs, researchers focus on bottleneck workstations of the fabs, such as the photolithography workstation (Graves *et al.* 1983, Lou and Kager 1989, Lee *et al*. 1995, Min and Yih 2003, Yoon and Lee 2004, and Lin *et al*. 2005). Batch scheduling problems have been

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

dealt with in a few studies as well. For example, Glassey and Weng (1991) give a method for scheduling jobs of a single job family on a single batch-processing machine, and Fowler *et al.* (1992), Robinson *et al.* (1995) and Fowler *et al.* (2000) deal with multi-product and multi-server cases.

Most of previous studies on production scheduling in semiconductor manufacturing systems have focused on objectives related to throughput, cycle time or equipment utilization, but due-date related performance measures have not been considered very often in those studies except for a few as surveyed in Fowler *et al.* (2002). Kim *et al.* (1998b, 2001) suggest dispatching-rule-based algorithms for lot release control and lot scheduling, and Kim *et al*. (2003) develop a real-time scheduling method in a wafer fab, for the objective of minimizing tardiness of orders. Also, Jain *et al*. (2003) develop a generalized stochastic Petri net model for wafer fabrication and suggest a scheduling strategy based on the simulated annealing method, and Mason *et al*. (2004) propose strategies for rescheduling jobs in complex job shops. Recently, Kim *et al.* (2008) develop a method for lot-order re-assignment under the soft pegging strategy, in which lot-order assignment can be changed during the manufacturing process.

Research on lot merging/splitting in semiconductor manufacturing systems is very rare. Liao *et al.* (1996) include a lot merging/splitting scheme in the formulation for scheduling problem in a semiconductor fabrication facility. However, they do not consider lot merging and splitting decisions for the development of a solution method. In this paper, we propose lot merging/splitting algorithms for the objective of minimizing the total tardiness of orders and cycle times of lots in a wafer fabrication facility that produces multiple product types.

2 LOT MERGING/SPLITTING ALGORITHM

Even though the product types of lots are different, lots can be merged into a single lot if the routes and the processing conditions of the lots are the same for a number of subsequent operations. Here, processing conditions denote those required for processing the operation, such as mask type, gas type, and processing temperature. We denote the wafer lots that can be merged together, i.e., those with the same processing conditions and the same next workstation to visit as a *wafer group*. (Therefore, merging can be done for the lots within the same wafer group.) If the routes or processing conditions become different in a later operation, the merged lot should be split into the original lots. Machines called *sorters* are used for merging or splitting those lots. The time required for merging or splitting is less than 5 minutes, which can be considered negligibly short compared to the processing time of operations, and hence it is not considered in this study.

Merging of lots can be done before processing of lots relevant to the merging decision is started. Therefore, a merging decision is made when one of the machines in a workstation becomes available and there are multiple partial lots of the same wafer group that can be processed on the machine. The merging decision is followed by a scheduling decision for determining a next lot to be processed on the machine. On the other hand, one does not have to consider splitting decisions, since merged lots are processed as a single lot until routes or processing conditions of the wafers included in the merged lots become different. That is, the splitting decision for a merged lot can be made automatically from the information on the routes or processing conditions for the wafers included in the merged lots.

In this study, we present several methods for determining whether partial lots waiting for a machine should be merged or not and how they should be merged. Before presenting these methods, we first describe scheduling methods to be used along with them. Basic concepts of these scheduling methods are also used in the methods for lot merging. In this study, it is assumed that the list scheduling method is used for scheduling in the fab as it is used practically in many fabs. In the list scheduling method, when a machine becomes available for processing a lot, a lot with the highest priority is selected among the lots that are available at the time and scheduled on the machine. One can reduce the cycle time or total tardiness by using good scheduling rules. We used scheduling rules given in Kim *et al*. (2001), ES/RW2 for lot scheduling at serial-processing workstations and MDBH for making batching and scheduling decisions, which showed good performance in terms of due-date related performance measures.

In ES/RW2, priorities of the lots are determined by estimated slack time per remaining work, and the operation due date is given to each operation considering its remaining work, not to each lot or order that includes the operation. In MDBH, when a (batch-processing) machine becomes available, wafer lots of a product family with the least average slack are selected for batching. If the number of the lots in the queue is greater than or equal to the batch capacity of a machine, a full batch is formed with the lots that have smaller slack values. Otherwise, two batching alternatives are considered: forming a batch with lots currently waiting in the queue and forming a batch with those lots and an additional lot that is expected to arrive first. Between two batching alternatives, the alternative with the least total weighted waiting time is selected and scheduled. See Kim *et al*. (2001) for more details of these rules.

In this study, we try to improve the productivity of a fab by reducing the number of lots that need to be processed through lot merging/splitting, since the total processing time is affected by the number of lots, especially on batch-processing machines, and generally it is easier to deal with a smaller number of processing units. We de-

velop four algorithms for merging partial lots that are waiting in a queue for being processed on a workstation.

Algorithm 1. First fit decreasing algorithm (FFD): This algorithm is adopted from the one used for bin-packing problems, the problems of packing a given set of items of different sizes into bins for a certain objective. Here, partial lots are regarded as items and merged lots are regarded as bins with the capacity of 25 wafers. In this algorithm, partial lots in a group are sorted in a non-increasing order of the sizes of the lots, and then according to this order, partial lots are included one by one in the lowest indexed bin (merged lot) with enough remaining capacity.

Algorithm 2. MDBH-FFD: This algorithm employs the basic concept used in MDBH (described above) of Kim *et al*. (2001). When a machine becomes available, wafer lots of a product family with the least average slack are selected for merging. If the number of wafers of the lots in the queue is greater than or equal to the capacity of a lot, those lots are merged into a new lot with algorithm FFD. Otherwise, two alternatives for lot merging are considered: merging lots currently waiting in the queue and merging those lots and an additional lot that is expected to arrive first. Between the two alternatives, the alternative that results in the least total weighted waiting time is selected for merging.

Algorithm 3. Knapsack-problem-based algorithm 1 (KS-1): Lots to be merged are selected by solving the following knapsack problem with the objective of minimizing the sum of waiting times of the lots. Only partial lots waiting in the queue are included in the knapsack problem.

$$\text{Maximize} \sum_{i=1}^{m} w_i x_i$$

$$\text{subject to} \sum_{i}^{m} a_i x_i \le b$$

$$x_i \in \{0, 1\} \text{ for all } i$$

Here, m is number of lots waiting in the queue, w_i is the waiting time of lot i, i.e., the interval between the time when lot i arrived at the workstation and the time when the merging decision is made, a_i is the size of (the number of wafers included in) lot i, b is the capacity of a lot, and x_i is the decision variable, which is equal to 1 if lot i is selected to be merged, and 0 otherwise.

Algorithm 4. Knapsack-problem-based algorithm 2 (KS-2): This algorithm is identical to KS-1 except for the objective function of the knapsack problem. In this algorithm, the objective function to be maximized is set to $\sum_{i=1}^{m} x_i / s_i$, where s_i is the ES/RW2 value of lot i, which is the value of the priority function used in the ES/RW2 rule for scheduling at the serial-processing workstation (Kim *et al.* 2003). The lot selected by ES/RW2 rule has less slack time and larger remaining work. That is, the ES/RW2 values can be considered to represent the urgencies of the lots, and hence in this algorithm, more urgent lots are merged together at earlier time.

After lots are merged into a single lot using the algorithms, the due date of the merged lot is set to the earliest due date of the lots included in the merged lot. As stated earlier, when a machine becomes available for processing, partial lots waiting for the machine are considered for possible merging and then a lot is selected for processing among full lots and merged lots as well as partial lots (that have not been merged) using the scheduling rules. Also, for scheduling decisions at serial-processing workstations, the ES/RW2 value of the merged lots is calculated as the minimum value of ES/RW2 of the lots included in the merged lot. Note that only when the merged lot is selected for being processed in the scheduling algorithm, the partial lots are actually merged and processed.

The overall procedure for lot merging and scheduling suggested in this study can be summarized as follows. This procedure is executed when a machine becomes available.

Procedure 1. (*Lot merging and scheduling*)

Step 1: Select a set of partial lots to be merged to a single lot by using one of the above four algorithms for each wafer group that can be processed on the machine. The lots selected to be included in merged lots are not actually merged yet at this point. If the machine is a serial-processing machine, go to step 2. Otherwise, (if the machine is a batch-processing machine) go to step 3.

Step 2: (for serial-processing machine) Select a lot with the minimum ES/RW2 value among all available full, merged or partial lots. Go to step 4.

Step 3: (for batch-processing machine) Apply the MDBH algorithm to all available full, merged or partial lots to make a batching and scheduling decision.

Step 4: If a merged lot is selected for processing in step 2 or if merged lots are included in the batch in step 3, perform merging the partial lots which are selected for merging in step 1, and process the merged lot in the machine. Otherwise, process the selected full lot or the selected partial lot, and the merging decision in step 1 is ignored. (No lots are merged in this case.)

3 SIMULATION EXPERIMENT

The performance of the lot merging/splitting algorithms suggested in this study is evaluated through simulation experiments. For the experiments, we generated problem instances based on data of a real fab in a semiconductor manufacturing company in Korea. The following summarize information of the real fab as well as wafers and orders used in the simulation model.

1. Eight workstations were included in the model: chemical/mechanical polishing, chemical vapor decomposition, diffusion, dry etching, implantation, photolithography, sputter, and cleaning, each with multiple parallel

machines. There are 501 machines, and 196 machines of them are batch processing machines.
2. There are 1100 different product types.
3. The size of (the number of wafers for) an order ranges from 25 to 300 wafers.
4. The processing time for a product (wafer) on a machine ranges from 5 to 240 minutes.
5. The number of operations required for a product ranges from 121 to 266.
6. Each product is composed of 10 to 15 layers of circuits, and hence, each wafer lot should visit workstations up to 10 to 15 times.
7. About 30% of the lots are partial lots.

In the simulation model, it is assumed that orders for approximately 3000 wafers arrive in each day, as in the real fab. The due date of order k, d_k, was given as

$$d_k = a_k + P_k \cdot \mathrm{TN}(2.184, 0.74^2; 1.1, \infty),$$

where a_k is the time when order k arrives, P_k is the sum of processing times of all operations for the order and TN(m,v;l,u) is a random number generated from a truncated normal distribution with mean m, variance v, lower limit l, and upper limit u. The lot sizes of the partial lots were randomly generated from the discrete uniform distribution with range, [1, 24].

As the method for releasing lots into the fab, a rule suggested in Kim *et al*. (2001), named the order slack rule (OS), was used for selection of a lot to be released, and the uniform release rule (UNIF), as given in Glassey and Resende (1988), was used for determining the time when the selected lot is to be released. In OS, a lot with the highest priority is released into the fab first. The priority of lot j is computed as

$$d'_j - Q_j - t - W_j - \beta \cdot N_j,$$

where d'_j is the due date of lot j, Q_j is the sum of processing times of all operations for lot j, W_j is an estimated total waiting time of lot j at the bottleneck station, N_j is the number of lots required for the order associated with lot j, and β is a parameter used in the release rule. In the simulation, β was set to 10 (after tests on several candidate values). Also, in UNIF, a selected lot is released into the fab in a constant rate (up to 3000 wafers a day) regardless of the current systems states. (Note that a rule with the same basic concept as that of UNIF is used in the fab considered in this study.) In the simulation experiments, we used scheduling rules given in Kim *et al*. (2001), ES/RW2 for lot scheduling at serial-processing workstations, and MDBH for scheduling at batch-processing workstations, since they showed good performance in terms of due-date related performance measures.

To evaluate the performance of the merging/splitting algorithms suggested in this research, we obtained benchmark solutions using the merging/splitting method used in the semiconductor wafer fab considered in this research. In the fab, the scheduling decision is made prior to the lot-merging decision. That is, when a machine becomes available, a lot is selected first with a scheduling rule. If the selected lot is a partial lot, another lot of the same wafer group as that of the selected lot is selected for merging in such a way that the size of the resulting merged lot becomes closest to 25 after merging. This algorithm will be denoted as REAL in this paper. To compare the performance of the merging methods only (not the scheduling rules), we use the same scheduling rule for all the merging algorithms/method. Therefore, ES/RW2 is used in REAL as well in the tests.

The tests include five merging algorithms/method, FFD, MDBH-FFD, KS-1, KS-2 and REAL. The simulation model was coded with Factor/AIM, a simulation software package developed by Pritsker Corporation, with additional user codes written in the C programming language. The simulation experiments were performed on a personal computer with a Pentium IV processor operating at 3.2GHz clock speed. In each simulation run, the period of 6 months was simulated and results of the last 5 months were used for comparison. We considered two scenarios corresponding to the ratio of partial lots to all lots. In the first scenario, which represents the current states of the fab, 30% of the lots are partial lots, while 40% of the lots are partial lots in the second scenario.

Table 1: Results of tests on scenario 1

Merging Algorithm	% reduction†		
	cycle time	Total tardiness	number of scheduled lots
REAL	4.3	0.8	0.5
FFD	8.1	2.9	3.5
MDBH-FFD	8.0	10.7	3.5
KS-1	9.5	24.9	3.4
KS-2	9.1	31.6	3.4

Table 2: Results of tests on scenario 2

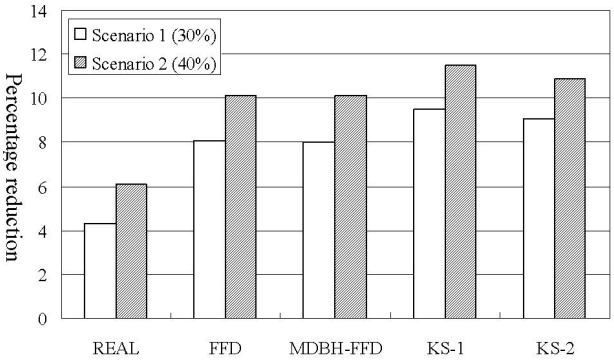
Merging Algorithm	% reduction†		
	cycle time	Total tardiness	number of scheduled lots
REAL	6.1	0.8	0.6
FFD	10.1	0.5	4.3
MDBH-FFD	10.1	11.5	4.3
KS-1	11.5	26.3	4.1
KS-2	10.9	33.8	4.2

Results of the simulation experiments are given in Tables 1 and 2, which show the percentage reduction of (average) cycle times of the lots and tardiness of the orders from the result of the case in which lot merging/spitting has not been done. It took about 2 hours for one simulation run. The

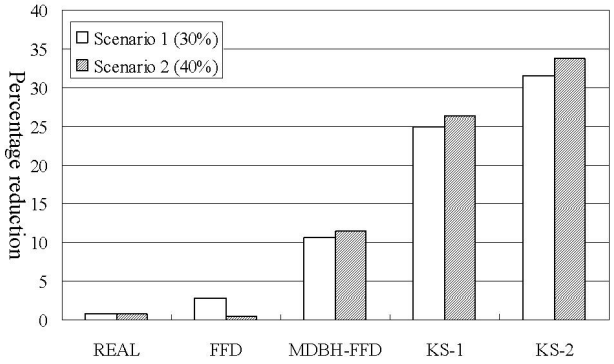
cycle time of a lot is estimated by the difference from the release time to the completion time of the lot regardless of lot merging/splitting during the processes. In addition, the tables also show the percentage reduction of the number of the lots, merged and original ones, that have been considered for scheduling from that should have been considered for scheduling without lot merging/spitting. Results for the two scenarios were almost the same. All the proposed algorithms showed better performance than REAL. This may be because more than two partial lots can be merged into one lot in the proposed algorithms, while no more than two partial lots can be merged into one in REAL.

Results on the cycle times of the lots were not much different for all the suggested merging algorithms, and neither were the numbers of the lots considered for scheduling. However, among the proposed algorithm, KS-1 showed a better performance than others in the measure of the cycle time of the lots. This may be because waiting times of lots in a queue are used in the objective function of the knap-sack problem, and hence lots that have waited for longer time in the queue tend to be merged together and processed earlier. On the other hand, the differences in the total tardiness of the orders among the proposed merging algorithms are more apparent. KS-2 worked best, followed by KS-1. Note that in KS-2 the slack time over remaining work is used in the objective function and hence more urgent lots, those with smaller value for this ratio, tend to be merged and processed earlier.

By applying the merging/splitting operations when needed or when possible, one can obtain better results. Also, the tables show that the merging algorithms suggested in this study outperformed the method currently used in the real fab in both performance measures, total tardiness and flow time (cycle time). In addition, comparison of the results of the two scenarios showed that when there are more partial lots, the outperformance of the suggested lot merging algorithms over the method currently used in the fab became clearer as can be seen from Figure 1. This may be because there are more lots that can be considered for merging and in such circumstances, effects of merging methods become more significant. However, the results for throughput were not much different for all the suggested merging algorithms and for the two scenarios. This may be because the fab produces wafer lots under the make-to-order policy, that is, wafer lots are produced up to the order quantities only even though more wafers can be produced in the fab.



(a) Percentage reduction of cycle times



(b) Percentage reduction of tardiness

Figure 1: Performance of the merging algorithms for different scenarios

4 CONCLUDING REMARKS

In this paper, we suggested lot merging algorithms to reduce the total tardiness of orders and the cycle times of lots in a semiconductor manufacturing fab facility. Results of comparison with a method currently used in real fab showed that a knapsack-problem-based algorithm showed the best performance among the algorithms for the measure of tardiness of orders, and another knapsack-problem-based algorithm worked best for the measure of cycle times of the lots. The results also show that by merging the lots with the same route and the same processing conditions, we can reduce not only the number of lots that need to be processed but also the cycle time of lots and total tardiness of orders can be reduced. As the effect of merging becomes more significant when there are more partial lots, the suggested merging algorithms may be more effective in cases in which there are more orders to be processed and/or there are more product types to be processed in a fab.

REFERENCES

Fowler, J., G. L. Hogg, and S. J. Mason. 2002. Workload control in the semiconductor industry. *Production Planning and Control* 13: 568-578.

Fowler, J., G. L. Hogg, and D. T. Phillips.1992. Control of multiporduct bulk service diffusion/oxidation processes. *IIE Transactions* 24: 84-96.

Fowler, J., G. L. Hogg, and D. T. Phillips. 2000. Control of multiproduct bulk server diffusion/oxidation processes. Part 2: Multiple severs. *IIE Transactions* 32: 167-176.

Glassey, C. R., and M. G. C. Resende. 1988. A scheduling rule for job release in semiconductor fabrication. *Operations Research Letters* 7: 213-217.

Glassey, C. R., and W. W. Weng. 1991. Dynamic batching heuristic for simultaneous processing. *IEEE Transactions on Semiconductor Manufacturing* 4: 77-82.

Graves, S. C., H. C. Meal, D. Stefek, and A. H. Zeghmi. 1983. Scheduling of re-entrant flow shops. *Journal of Operations Management* 3: 197-207.

Jain, V., R. Swarnkar, and M. K. Tiwari. 2003. Modelling and analysis of wafer fabrication scheduling via generalized stochastic Petri net and simulated annealing. *International Journal of Production Research* 41: 3501-3527.

Kim, Y.-D., J.-Y. Bang, K.-Y. An, and S.-K. Lim. 2008. A due-date based algorithm for lot-order assignment in a semiconductor wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing* 21: 209-216.

Kim, Y.-D., J.-U. Kim, S.-K. Lim, and H.-B. Jun. 1998a. Due-date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing* 11: 155-164.

Kim,Y.-D., J.-G, Kim, B. Choi, and H.-U. Kim. 2001. Production scheduling in a semiconductor wafer fabrication facility producing multiple product types with distinct due dates. *IEEE Transactions on Robotics and Automation* 17: 589-598.

Kim, Y.-D., D.-H. Lee, J.-U. Kim, and H.-K. Roh. 1998b. A simulation study on lot release control, mask scheduling, and batch scheduling in semiconductor wafer fabrication facilities. *Journal of Manufacturing Systems* 17: 107-117.

Kim,Y.-D., S.-O. Shim, B. Choi, and H. Hwang. 2003. Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing* 16: 290-298.

Lee, K.-H., Y. Hong, and S.-Y. Kim. 1995. Production scheduling in semiconductor wafer fabrication process. *Journal of the Korean Institute of Industrial Engineers* 21: 357-369.

Lin, J. T., F. K. Wang, and P. C. Kuo. 2005. A parameterized-dispatching rule for a logic IC sort in a wafer fabrication. *Production Planning and Control* 16: 426-436.

Liao, D.-Y., S.-C. Chang, K.-W. Pei, and C.-M. Chang. 1996. Daily scheduling for R&D semiconductor fabrication. *IEEE Transactions on Semiconductor Manufacturing* 9: 550-561.

Lou, S. X. C., and P. W. Karger. 1989. A robust production control policy for VLSI wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing* 2: 159-164.

Mason, S.J., S. Jin, and C. M. Wessels. 2004. Rescheduling strategies for minimizing total weighted tardiness in complex job shops. *International Journal of Production Research* 42: 613-628.

Min, H. S., and Y. Yih. 2003. Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system. *International Journal of Production Research* 41: 3921-3941.

Robinson, J. K., J. Fowler, and J. F. Bard. 1995. Use of upstream and downstream information in scheduling semiconductor batch operations. *International Journal of Production Research* 33: 1849-1869.

Wein, L. M.. 1998. Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing* 1: 115-130.

Yoon, H. J., and D. Y. Lee. 2004. Deadlock-free scheduling of photolithography equipment in semiconductor fabrication. *IEEE Transactions on Semiconductor Manufacturing* 17: 42-54.

AUTHOR BIOGRAPHIES

JUNE-YOUNG BANG received a B.S. and an M.S. degrees in Mechanical Engineering from KAIST. He is a Ph.D. student in the Department of Industrial Engineering, KAIST. His research areas include operations scheduling and production control.

JAE-HUN KANG received a B.S. degree in Chemical Engineering from SNU. He is a Ph.D. student in the Department of Industrial Engineering, KAIST. His research areas include operations scheduling and production control as well as supply chain management.

BONG-KYUN KIM received a B.A. in English from Korea Military Academy and received a M.B.A from KDI School of Public Policy and Management. He is a Ph.D. student in the Department of Industrial Engineering, KAIST. His research areas include scheduling on military operations and production control.

YEONG-DAE KIM is a professor at the Department of Industrial Engineering, Korea Advanced Institute of Science and Technology (KAIST). He received a B.S. degree from Seoul National University, an M.S. degree from KAIST both in Industrial Engineering, and a Ph.D. degree from the University of Michigan in Industrial and Operations Engineering. His research areas include design and operation of manufacturing systems, operations scheduling and production planning and inventory management. He is a senior member of IIE, a member of INFORMS, the Operational Research Society and IEEE.