Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds..

IMPROVING THE PERFORMANCE OF DISPATCHING RULES IN SEMICONDUCTOR MANUFACTURING BY ITERATIVE SIMULATION

Lars Mönch

Institute of Information Systems
Technical University of Ilmenau
Helmholtzplatz 3, P.O. BOX 100565
D- 98684 Ilmenau, GERMANY

Jens Zimmermann

Department of System Analysis
Technical University of Ilmenau
Gustav-Kirchhoff-Straße 1, P.O. BOX 100565
D- 98684 Ilmenau, GERMANY

ABSTRACT

In this paper, we consider semiconductor manufacturing processes that can be characterized by a diverse product mix, heterogeneous parallel machines, sequence-dependent setup times, a mix of different process types, i.e. single-wafer vs. batch processes, and reentrant process flows. We use dispatching rules that require the estimation of waiting times of the jobs. Based on the lead time iteration concept of Vepsalainen and Morton (1988), we obtain good waiting time estimates by using exponential smoothing techniques. We describe a database-driven architecture that allows for an efficient implementation of the suggested approach. We present results of computational experiments for reference models of semiconductor wafer fabrication facilities. The results demonstrate that the suggested approach leads to high quality solutions.

1 INTRODUCTION

The manufacturing of integrated circuits on silicon wafers is a manufacturing process of huge complexity (Uzsoy, Lee, and Martin-Vega 1992, Schömig and Fowler 2000). Semiconductor wafer fabrication facilities (wafer fabs) are characterized by heterogeneous parallel machines, a mix of different process types, i.e. batch processes vs. single wafer processes, reentrant process flows and prescribed customer due dates of the orders.

Currently it seems that the improvement of operational processes creates the best opportunity to reduce costs inside the wafer fabs (Schömig and Fowler 2000). The new information technology opportunities have to be taken into account during the development of new planning and control strategies.

Dispatching schemes are still the major tool used for production control in wafer fabs. Dispatching rules provide a very quick solution, but they are myopic in time and space, i.e. usually they use only local information for decision-making. Global dispatching rules take more global information into account in order to calculate the priority index. In this paper, we study a scheme that allows for a configuration of a certain class of global dispatching rules. The global information are given by waiting time estimates for single process steps for each released job.

An iterative simulation-based scheme was suggested by Vepsalainen and Morton (1988) in order to come up with waiting time estimates. However, so far only little is known on the performance of this scheme under complex process conditions as in semiconductor manufacturing. We study the behavior, suggest modifications of the original scheme and describe a modern software architecture that can be used for the implementation of the algorithms.

The paper is organized as follows. In the next section, we describe the considered problem. We continue with presenting our solution approach. Then we show the results of computational experiments with two different wafer fab simulation models.

2 PROBLEM SETTING AND NOTATION

We study global dispatching rules that take waiting times into account. The waiting times for process steps that have to be performed in the future are unknown. The waiting times depend, for example, on the product mix, the load of the wafer fab and on the used control strategy.

Global variants of the Apparent Tardiness Cost (ATC) dispatching rule (Vepsalainen and Morton 1987) are discussed in Vepsalainen and Morton (1988). The authors suggest a solution that is based on iterative simulation called lead time iteration. Based on a crude initial waiting time estimate, successive adjustments of the waiting times are performed by using the measured waiting times from the current simulation run.

This method was also used by Ovacik and Uzsoy (1997) in order to determine appropriate internal due dates for an operational due date type dispatching rule in the test area of a semiconductor wafer fab.

In (Lu, Ramasawamy, and Kumar 1994) the authors apply a lead time iteration scheme in order to estimate waiting times for a certain class of dispatching rules called Fluctuation Smoothing Policy.

In this study, we consider a global ATCS rule (cf. Vepsalainen and Morton 1988). The index has to be calculated as follows.

$$I_{ij,ATCS}(t,lk) := \frac{w_j}{p_{ij}} exp\left(-\frac{s_{ij}^+}{\kappa_1 \bar{p}}\right) exp\left(-\frac{s_{lk,ij}}{\kappa_2 \bar{s}}\right), \quad (1)$$

where we set for abbreviation

$$s_{ij} := d_i - t - p_{ij} - \sum_{g=j+1}^{n_i} \left(w_{ig} + p_{ig}\right). \quad (2)$$

Here, we use the following notation:

ij :	process step j of job i,
t :	time for decision-making,
d_i :	due date of job i,
p_{ij} :	processing time of step j of job i,
w_i :	weight of job i,
κ_1 :	scaling parameter for the slack term s_{ij},
κ_2 :	scaling parameter for the set-up term,
$\bar{p}$:	average processing time of the remaining jobs,
$\bar{s}$:	average set-up time of the remaining jobs,
$s_{lk,ij}$:	set-up time that occurs if the process step j of job i is processed after process step k of job l,
w_{ij} :	waiting time for processing process step j of job i,
n_i :	number of process steps of job i.

Furthermore, we will use $x^+ := max(x,0)$ throughout the rest of the paper. The first term in expression (1) represents the index of the weighted shortest processing time rule, the second term takes the slack of the job into account, whereas the third term is used to model the impact of sequence-dependent set-up times.

In the case of batching tools, we calculate the priority index of a certain batch by taking the sum over the ATCS indices given by formula (1). In order to form a certain batch, we first sequence the jobs of one incompatible family according to the ATCS index (1) in a non-increasing manner. Then we take the first B jobs in order to form the (full) batch. We select the batch with the highest sum of the ATCS indices among the incompatible job families for processing next. Here, the quantity B denotes the maximum batch size, i.e. the capacity of a batch machine.

We denote the flow factor by FF. The flow factor is the ratio of the average cycle time and the raw processing time.

We consider a second dispatching rule that is called slack per remaining processing time (SRPT). The corresponding priority index has to be calculated as follows:

$$I_{ij,SRPT}(t,lk) := \frac{s_{ij}}{\sum_{g=j+1}^{n_i} p_{ig}} exp\left(\frac{s_{lk,ij}}{\kappa_2 \bar{s}}\right). \quad (3)$$

Here, we use the same notation as for the description of the ATCS index. For batch formation issues we proceed as in the case of the ATCS rule.

We use the COVERT rule as a third dispatching scheme in our experiments. The corresponding priority index is defined as follows:

$$I_{ij,COVERT}(t,lk) := \left\{1 - \frac{sq_{ij}^+}{h\sum_{g=j}^{n_i} w_{ig}}\right\}^+ exp\left(-\frac{s_{lk,ij}}{\kappa_2 \bar{s}}\right), \quad (4)$$

where we denote by

$$sq_{ij} := d_i - t - \sum_{g=j}^{n_i} p_{ig} \quad (5)$$

a slack-type quantity. The quantity h in expression (5) is used for scaling purposes. For batch formation issues we proceed again as in the case of the ATCS rule.

3 SOLUTION APPROACH

We describe first the lead time iteration approach. The used software architecture is discussed in the second part of this section.

3.1 Lead Time Iteration

The Lead Time Iteration (LTI-I) procedure can be formulated for job i as follows:

1. Get an initial waiting time estimate using the flow factor concept, i.e. set

$$w_{ij}^{(0)} := (FF - 1)\, p_{ij}, \quad (6)$$

for the waiting time connected with process step ij.

2. Dispatch the wafer fab using the ATCS, SRPT or COVERT dispatching rule.
3. Calculate the actual waiting time q_{ij} of each process step *ij* from simulation run *l*. Here, the waiting time is defined as the time between the completion of process step *ij*-1 and the start time of process step *ij*
4. Update the waiting time estimate as follows:

$$w_{ij}^{(l+1)} := (1-\alpha) w_{ij}^{(l)} + \alpha \; q_{ij}, \tag{7}$$

where $\alpha \in (0,1)$ denotes a fixed smoothing factor.

5. Terminate the iterative procedure if a stopping condition as described in (12) is valid, otherwise go to Step 2.

We apply second order exponential smoothing (cf. Tempelmeier 1992) in order to obtain a faster convergence of the iterative simulation scheme. Second order exponential smoothing models the trend. We call the suggested scheme LTI-II. It can be described as follows:

1. Get initial waiting time $w_{ij,1}^{(0)}, w_{ij,2}^{(0)}$ estimates using the flow factor concept, i.e. set

$$w_{ij,1}^{(0)} = w_{ij,2}^{(0)} = (FF-1)\, p_{ij}. \tag{8}$$

2. Dispatch the wafer fab using the ATCS, SRPT, or COVERT dispatching rule.
3. Determine the actual waiting time q_{ij} of each process step *ij* from the simulation run.
4. Update the waiting time estimate as follows:

$$w_{ij,1}^{(l)} := (1-\alpha) w_{ij,1}^{(l-1)} + \alpha \; q_{ij}, \tag{9}$$

$$w_{ij,2}^{(l)} := (1-\alpha) w_{ij,2}^{(l-1)} + \alpha \; w_{ij,1}^{(l)}, \tag{10}$$

$$w_{ij}^{(l+1)} := 2w_{ij,1}^{(l)} - w_{ij,2}^{(l)} + \frac{\alpha}{1-\alpha}\left(w_{ij,1}^{(l)} - w_{ij,2}^{(l)}\right). \tag{11}$$

Here, the quantities $w_{ijl}^{(l)}$ have an immediate character and are used only for determining the final waiting time estimate $w_{ij}^{(l+1)}$ for iteration *l*+*1*.

5. Terminate the iterative procedure if a stopping condition as described in (12) is valid, otherwise go to Step 2.

We terminate both algorithms if the following condition holds for the first time:

$$\max_{ij} \left| w_{ij}^{(l)} - w_{ij}^{(l+1)} \right| \le \varepsilon, \tag{12}$$

where ε denotes a prescribed threshold value. Usually four to eight iterations are enough for LIT-I. In the case of LTI-II we have to perform only five iterations.

The waiting time updates in Step 4 of the LTI-I and LTI-II algorithms take the current measured waiting time from the simulation run and the estimated waiting time from the previous iteration into account.

3.2 Software Architecture

The used software architecture based on a generic architecture for simulation-based performance assessment of shop-floor control systems (Mönch, Rose, and Sturm 2003). The main ingredient of the architecture is a blackboard-type data-layer between the control program and the simulation engine. The blackboard contains objects that are similar to those in the simulation engine in a mirror-like fashion.

Simulation events are used for updating the objects of the data-layer. The control program calls the simulation engine after each single iteration and calculates waiting times in each single iteration of the algorithm.

We extend the basic architecture suggested by Mönch, Rose, and Sturm (2003) by an object-oriented database. Here, we can store the waiting time for each single process step in each iteration. The database makes the waiting times persistent. An object-oriented database is appropriate, because we can easily store nested objects. In the our case, such nested objects are given by the structure of the routes and by the assignment of waiting times to process steps of the jobs. We show the suggested architecture in Figure 1.

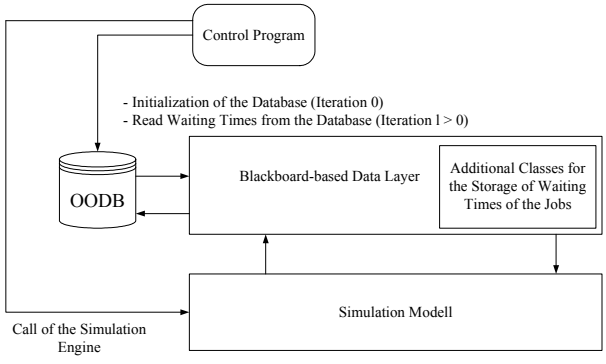

Figure 1: Software Architecture for Iterative Simulation

We use the simulation engine AutoSched AP 7.1 and the database FastObjects in order to develop our scheduling application. The control application is implemented in the C++ programming language.

4 COMPUTAIONAL RESULTS

We consider two different simulation models. Model A (cf. El Adl, Rodriguez, and Tsakalis 1996) contains five workstations that are organized into three tool groups. One tool group has batching characteristic, whereas a second one requires sequence-dependent set-up times. We use two products, each of them includes seven process steps.

Model B is the MIMAC test data set 1 (Fowler and Robinson 1995). It contains over 200 tools that are organized into over 80 tool groups. Two product routes are included into the model. Each route contains more than 200 process steps.

We expect that the following properties that are related to the manufacturing system influence the performance of the iterative simulation scheme:

- Due date of the jobs. Our due date setting is based on the flow factor concept. We calculate the due date of job *i* by

$$d_i := r_i + FF\sum_{j=1}^{n_i} p_{ij} \ , \tag{13}$$

  where we denote by r_i the ready time of job *i*.
- Weighting scheme for the jobs,
- Load of the wafer fab.

Beside the characteristics of the manufacturing system, certain parameters of the iterative simulation procedure are important with respect to the performance of the algorithm. The parameters are:

- Order of exponential smoothing, i.e. using the LTI-I or LTI-II scheme,
- Smoothing factor α.

We consider two different weight distributions for the jobs. The distribution D_1 is defined as follows:

$$D_1 := \begin{cases} w_i = 1 & with \quad p_1 = 0.50, \\ w_i = 5 & with \quad p_2 = 0.35, \\ w_i = 10 & with \quad p_3 = 0.15. \end{cases} \tag{14}$$

Distribution D_1 mimics the situation that a small number of jobs have a high weight and a large number of jobs have a medium weight. The distribution D_2 is given by

$$D_2 := \begin{cases} w_i = 1 & with \quad p_1 = 0.50, \\ w_i = 2 & with \quad p_2 = 0.45, \\ w_i = 10 & with \quad p_3 = 0.05. \end{cases} \tag{15}$$

The second distribution is used to model manufacturing systems where a very small portion of the jobs have a high priority and the remaining jobs have a small weight.

We summarize the resulting factorial design used for the experiments in Table 1.

Table 1: Factorial Design for Model A and B

Factor	Level	Count
Order of Exponential Smoothing	1;2	2
Smoothing Factor	0.7;0.9	2
Load of the Wafer Fab	Low;High	2
Due Dates	*FF=1.6; FF= 2.0;*	2
Weights	$D_1; D_2$	2
Stochastically Independent Samples of the Weights	Five Samples for both $D_1; D_2$	10
Stochastically Independent Simulation Replications	-	5

We show results of simulation experiments for model B in Table 2. All results are presented as ratio of the performance measure value obtained by the final iteration of the iterative simulation scheme and of the performance measure value obtained by using the First In First Out (FIFO) dispatching rule.

Table 2: Results for LTI-I and LTI-II for ATCS for Model B

	TWT	**CT**	**TP**
Order of Exponential Smoothing			
1	0.4963	0.9435	1.0075
2	0.5018	0.9447	1.0075
Smoothing Factor			
$\alpha = 0.7$	0.4988	0.9438	1.0075
$\alpha = 0.9$	0.4992	0.9443	1.0073
Load			
Low	0.5550	0.9583	1.0064
High	0.4431	0.9298	1.0085
Due Date Setting			
$FF = 1.6$	0.4357	0.9446	1.0073
$FF = 2.2$	0.5623	0.9435	1.0076
Weighting Scheme			
D_1	0.4385	0.9461	1.0068
D_2	0.5595	0.9420	1.0081
Average	0.4990	0.9441	1.0075

The results from Table 2 have to be interpreted in the following way. A special factor value in the first column means that only the average value for those experi-

ments is reported where the factor has the specific value from the column.

By TWT we denote the total weighted tardiness of the jobs defined as

$$TWT := \sum_{i=1}^{n} w_i (c_i - d_i)^+ , \qquad (16)$$

where we denote the completion time of job *i* by c_i and by n the number of completed jobs. The second performance measure is the (average) cycle time (CT). It is defined as

$$CT := \frac{1}{n} \sum_{i=1}^{n} (c_i - r_i), \qquad (17)$$

where we denote by r_i the ready time of job *i*. The third performance measure is the throughput (TP) measured in jobs. It is the number of completed jobs within a certain time period.

We consider a simulation time of 180 days. We start from an appropriate WIP distribution in order to reduce the required warm-up period. We used those values for κ_1 and κ_2 that lead to a minimum TWT. In order to determine these values, we consider 8^2 simulation runs for pairs $(\kappa_1, \kappa_2) \in [0.1;6.0] \times [0.1;8.0]$.

From Table 2, we conclude that the iterative simulation scheme leads to significant total weighted tardiness reductions for each single factor. As expected, the improvement rate is higher for a wafer fab with high load. Furthermore, in the case of tight due dates (*FF=1.6*) the improvement rate is also higher. A smaller smoothing factor α leads to slightly better results. We present the corresponding results for the dispatching scheme SRPT in Table 3.

Table 3: Results for LTI-I and LTI-II for SRPT for Model B

	TWT	CT	TP
Order of Exponential Smoothing			
1	0.7532	0.9814	1.0037
2	0.7531	0.9842	1.0037
Smoothing Factor			
$\alpha = 0.7$	0.7609	0.9837	1.0031
$\alpha = 0.9$	0.7454	0.9819	1.0043
Load			
Low	0.7798	0.9816	1.0054
High	0.7266	0.9841	1.0020
Due Date Setting			
FF = 1.6	0.9652	0.9945	1.0010
FF = 2.2	0.5411	0.9712	1.0064
Weighting Scheme			
D_1	0.7243	0.9828	1.0037
D_2	0.7820	0.9828	1.0037
Average	0.7532	0.9828	1.0037

It turns out that we obtain similar results as in the case of the ATCS dispatching rule. However, the ATCS rule performs better than the SRPT rule. We obtain a similar behavior for the COVERT rule.

For model comparison purposes, we present also results for ATCS applied to model A in Table 4. We see that we get results of similar quality. However, as expected the magnitude of improvements is lower compared to model B. Model B offers much more room for improvement because of its complexity.

Table 4: Results for LTI-I and LTI-II for ATCS for Model A

	TWT	CT	TP
Order of Exponential Smoothing			
1	0.8656	0.9748	1.0002
2	0.8772	0.9755	1.0002
Smoothing Factor			
$\alpha = 0.7$	0.98538	0.9751	1.0002
$\alpha = 0.9$	0.8890	0.9751	1.0002
Load			
Low	0.9862	0.9916	1.0007
High	0.7566	0.9586	0.9996
Due Date Setting			
FF = 1.6	0.8177	0.9749	1.0002
FF = 2.2	0.9250	0.9753	1.0002
Weighting Scheme			
D_1	0.8626	0.9758	1.0002
D_2	0.8801	0.9744	1.0002
Average	0.8845	0.9751	1.0002

In Figure 2, we present the typically obtained TWT values after a certain number of iterations.

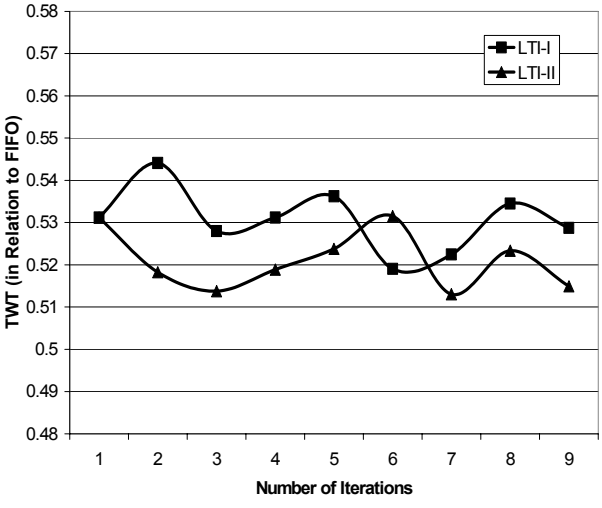

Figure 2: Convergence Speed for LTI-I and LTI-II

We see that in case of LTI-II a smaller number of iterations is enough. Here, we present the TWT values relative to the TWT values obtained by a FIFO dispatching policy.

We see from Figure 3 that the convergence of algorithm LTI-II compared to algorithm LTI-I is in a large number of scenarios faster. Here, convergence speed is measured by the number of iterations that are required in order to get the best result with respect to TWT.

On the other hand, we obtain from Table 2, Table 3, and Table 4 that the performance of LTI-II is only slightly poorer as the performance of LIT-I with respect to TWT.

In Figure 3, we consider totally 32 different scenarios according to the factorial design from Table 1.

5 CONCLUSIONS

In this paper, we discuss modifications of the lead time iteration scheme of Morton and Vepsalainen. We present a software architecture that allows for an efficient implementation of our algorithm. We perform simulation experiments that show the high quality of the obtained solutions in many situations.

We are interested to integrate our approach into a scheduling engine. Based on the waiting time estimates it seems to be possible to calculate schedules for the entire wafer fab. Using these schedules, an improvement of certain due date oriented performance measures seems to be possible. Hence, the scheme could be embedded into a more general simulation-based scheduling framework.

ACKNOWLEDGMENTS

The authors would like to thank Torsten Michael for his valuable programming and simulation efforts.

REFERENCES

El Adl, M. K., A. A. Rodriguez, and K. S. Tsakalis. 1996. Hierarchical Modelling and Control of Re-entrant Semiconductor Manufacturing Facilities. *Proceedings of the 35th Conference on Decision and Control*. Kobe, Japan.

Fowler, J. W. and J. Robinson. 1995. Measurement and Improvement of Manufacturing Capacities (MIMAC): Final Report. Technical Report 95062861A-TR, SEMATECH, Austin, TX.

Lu, S. C. H., D. Ramaswamy, and P.R. Kumar. 1994. Efficient Scheduling Policies to Reduce Mean and Variance of Cycle Time in Semiconductor Manufacturing Plants. *IEEE Transactions on Semiconductor Manufacturing,* 7(3), 374-388.

Mönch, L., O. Rose, and R. Sturm. 2003. Simulation Framework for the Performance Assessment of Shop-Floor Control Systems. *SIMULATION: Transactions of the Society for Modeling and Simulation International*, 79(3), 163-170.

Ovacik. I. M. and R. Uzsoy. 1997. *Decomposition Methods for Complex Factory Scheduling Problems*. Kluwer Academic Publishers, Boston.

Schömig, A. and J. W. Fowler. 2000. Modeling Semiconductor Manufacturing Operations. In *Proceedings of the 9th ASIM Dedicated Conference Simulation in Production and Logistics*, ed. K. Mertins and M. Rabe, 55-64.

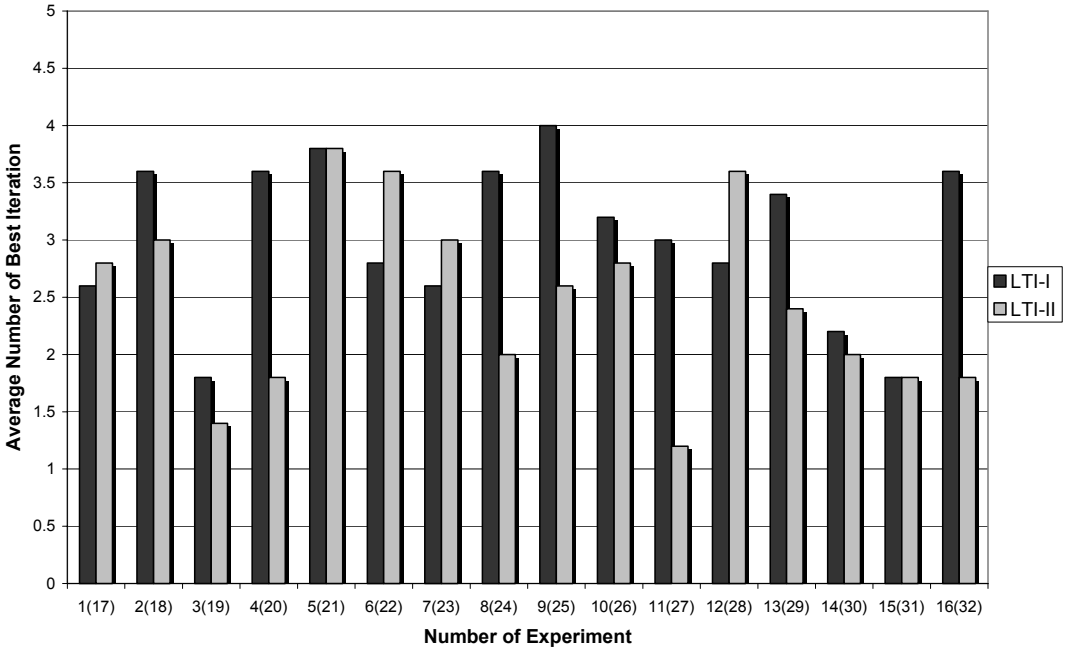

Figure 3: Number of the Iteration that Leads to Minimum TWT

Tempelmeier, H. 1992. *Material-Logistik: Grundlagen der Bedarfs- und Losgrößenplanung in PPS-Systemen*. Springer Verlag, Berlin.

Uzsoy, R., C.-Y. Lee, and L. A. Martin-Vega. 1992. A Review of Production Planning and Scheduling Models in the Semiconductor Industry, Part I: System Characteristics, Performance Evaluation and Production Planning. *IIE Transactions on Scheduling and Logistics*, 24, 47-61.

Vepsalainen, A. and T. E. Morton. 1987. Priority Rules and Lead Time Estimate for Job Shop Scheduling with Weighted Tardiness Costs. *Management Science*, 33, 1036-1047.

Vepsalainen, A. and T. E. Morton. 1988. Improving Local Priority Rules with global lead time Estimates: a Simulation Study. *Journal of Manufacturing and Operations Management* 1, 102-118.

AUTHOR BIOGRAPHIES

LARS MÖNCH is an Assistant Professor in the Department of Information Systems at the Technical University of Ilmenau, Germany. He received a master's degree in applied mathematics and a Ph.D. in the same subject from the University of Göttingen, Germany. His current research interests are in simulation-based production control of semiconductor wafer fabrication facilities, applied optimization and artificial intelligence applications in manufacturing. He is a member of GI (German Chapter of the ACM), GOR (German Operations Research Society), SCS and INFORMS. His email address is <Lars.Moench@tu-ilmenau.de>.

JENS ZIMMERMANN is a Ph.D. student in the Department of System Analysis at the Technical University of Ilmenau, Germany. He received a master's degree in information systems from the Technical University of Ilmenau. He is interested in semiconductor manufacturing, simulation and machine learning. He is a member of GI. His email address is <Jens.Zimmermann@tu-ilmenau.de>.