Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

SIMULATION-BASED PERFORMANCE ASSESSMENT OF AN IMPLANT SCHEDULER IN SEMICONDUCTOR MANUFACTURING

Thomas Winkler
Ralf Sprenger

GLOBALFOUNDRIES Dresden
Module One LLC & Co. KG
Wilschdorfer Landstrasse 101
Dresden, 01109, GERMANY

ABSTRACT

The processes of implanting tools in semiconductor production involve a variety of setup parameters, suggesting utilization of a scheduler for planning lot allocation in a reasonable manner. However, a scheduler is complex by design, making evaluation difficult without appropriate aids. We describe an approach using discrete event simulation software for assessing the performance and supporting the parameterization of an implant scheduler. Furthermore, rolling-horizon approaches are discussed and the scheduler performance is compared to two dispatch rules.

1 INTRODUCTION

Semiconductor manufacturing depends on hundreds of complex process and measurement steps to be passed by every single wafer. Different products and technologies require their distinct flows and machine setups. Especially in foundry business, where a steadily growing number of products and platforms can be observed, it is challenging to plan material (lots) and tools in an efficient way.

The processes of some tool groups – especially in the area of photolithography and ion implantation – are subject to special constraints. Ion implanters have to load the appropriate ion source (e. g. Germanium or Phosphor); apart from that, the ion energy, dose and acceleration voltage have to be setup accordingly to result in the desired density and profile of implanted ions. Ion sources have a defined maximum allowed runtime after which they must be changed. Some transitions between ion sources show negative effects and should be avoided, or the tool must be conditioned by supplying an inert species before loading the new source.

To minimize the amount of setup changes, maximize the usage of ion sources and satisfy species change constraints while not overly contradicting lot priority, material process order must be carefully planned for the tools. The described approach considers the existing scheduler as a black-box system and links it to a discrete event semiconductor fab simulation. This allows local and global objective or KPI evaluation under fairly realistic conditions, without incurring the risks of evaluating effects in a real fab. In Mönch, Rose, and Sturm (2003) and also in Mönch (2007) similar problems are discussed and a framework is proposed to evaluate the performance of Shop-Floor Control Systems. However, we use a deployed scheduler as is and evaluate its performance against simple dispatch rules.

In Section 2, we will discuss the problem in detail, elaborating on the necessary simulation model enhancements to support the Implant Scheduler specifics. Section 3 describes the linkage of the scheduler to the simulation. In the following we will discuss two simple dispatching logics developed as a reference to compare the scheduler against them. Section 5 will show results of the linked scheduler. The last

978-1-5386-3428-8/17/$31.00 ©2017 IEEE

section will summarize the findings and discuss reusability and future extensibility of the approach. Literature review is done within the single sections.

2 IMPLANT-SPECIFIC CONSTRAINTS

We start by specifying the constraints that need to be considered for modeling of ion implantation. Besides common processing characteristics like

- batch, cascade or single wafer process,
- process and interval times,
- number of chambers and/or load ports,
- scheduled and unscheduled down times, and
- setup changes,

that you will find at many tool types in semiconductor manufacturing, ion implantation processes require the additional special parameters described in Tables 1 and 2.

Table 1: Species, energy and dose constraints.

Implant constraint	**Description**
Species	Type of ions to implant, e.g. B, Al, Ga, In, P, As, Sb.
Energy	Acceleration energy in keV. Defines the penetration depth of the ions.
Dose	Concentration of implanted ions in cm^{-2}. The higher the dose, the denser and more amorphous the implantation.

Table 2: Species specific constraints.

Species constraint	**Description**
Species changes from → to not allowed	Hard constraint. If species *from* is loaded, never change directly to species *to*, e.g. because of high contamination risk.
Species changes from → to not desired	Soft constraint, disregarding it might impact ion source lifetime or yield.
Maximum species runtime	Species must be changed no later than after it has been loaded for the defined duration.
Minimum species runtime	Soft constraint to avoid expensive short runtimes (ensures it is worth it loading the species).
Inert time after unloading species	Minimum time a specified inert gas (often Argon) must be loaded in the tool to clean the tool from remains of the unloaded species before loading a new species.
Species re-reservation time	Minimum time before the same species can be reserved again after having been unloaded.
Used inert species	For cleaning and for idling, pausing the loaded species runtime.

Detailed discussion about implant related constraints can be found in Anders (2000). The discrete event simulation model has been enhanced to support reading and passing these additional parameters and constraints to the implant scheduler. Tracking functionality for detecting constraint violations has also been implemented into the simulation. This is useful because it enables to check if the scheduler makes decisions that do not violate any of the constraints. As a result, we found bugs within our production scheduler that was already running for a long time and were able to solve them.

3 LINKAGE OF SCHEDULER AND SIMULATION

Using a scheduler in a real fab leads to some execution specifics that need to be considered. A rolling-horizon approach is required because schedule calculation takes time and the fab situation will change due to stochastic events. In the following we will discuss which approaches can be used, how we implemented them in the simulation engine and the connection between scheduler and simulation.

3.1 Rolling-horizon approach

The scheduler computes lot schedules based on a given problem set, but data collection and computation themselves consume time during which production continues. Figure 1 shows the scheduler calculation and schedule execution. The scheduler is called every *m* minutes and computation takes *n* minutes. To model this behaviour within the simulation, we stop it during the time the scheduler is running but we delay its execution by the runtime of the scheduler. Some discussions about rolling-horizon approaches can be found in Sprenger and Mönch (2014) and Zang (2016).

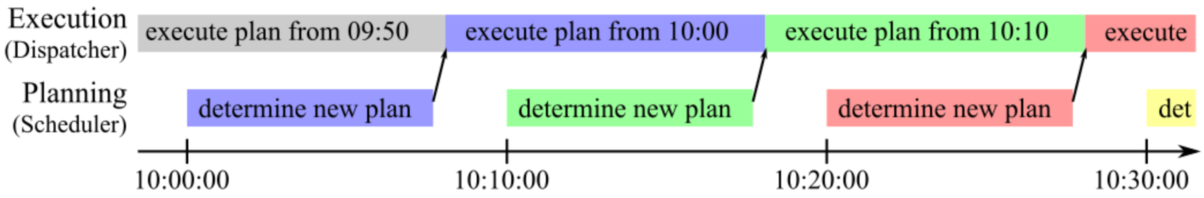

Figure 1: Example of rolling-horizon based planning with *m*=10 minutes, *n*=8 minutes.

3.2 Integration of simulation

The interfaces of the provided scheduler are designed to integrate into a real fab's Real-time Dispatching (RTD) system. For this simulation-based evaluation, we implemented an emulation of the dispatching system's relevant interfaces (call scheduler, provide lot data, accept schedule) into the discrete event simulation. This approach had the advantage that the scheduler could be used as-is (black-box approach), making it possible to test as comprehensive as possible and as closely as possible to a real fab environment.

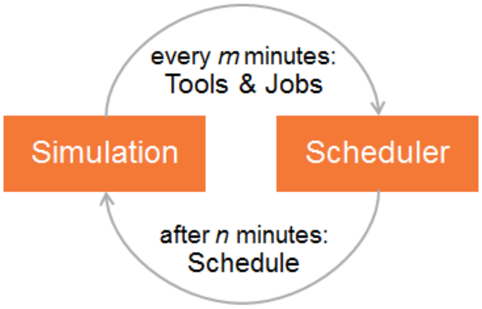

Figure 2: Schematic of alternating execution of simulation and scheduler.

Besides reading, tracking and delivering the special parameters and constraints described in the previous sections, the following extensions are necessary to run the scheduler from the simulation:

- Create periodic scheduling events every *m* minutes of simulation time, interrupting the simulation and calling the scheduler.
- Forecast of upcoming material for creating the scheduler input.
- Integration of schedule execution into simulation.

3.3 Forecast of upcoming material

Forecasting upcoming material is non-trivial and its quality becomes the more important the larger a *m* is chosen. Two different methods were implemented:

- **Method 1** considers all lots (1) waiting at an Implant operation, (2) being in transport towards an implant operation, or (3) processing at an operation immediately preceding to an implant operation. This method can predict upcoming material only up to *process time of predecessor operation + transportation time*. This makes it suitable only for a small *m*, as otherwise not enough upcoming material will be fed into the scheduler. However, this method is very fast.
- **Method 2** is based on the authors' previous work (Winkler, Barthel and Sprenger 2016) where we presented a method of deep copying the simulation model and running it for a given simulation time horizon *fh*, feeding back results – in this case the lots arriving at implant operations – of this copied simulation model to the main simulation run. By choosing an *fh* > *m*, we ensure that enough input lots will be given to the scheduler even for a large *m*. On the downside, this approach is comparatively slow, yet in relation to the scheduler's execution time still neglect able.

3.4 Managing uncertainty during schedule execution

A real fab is highly stochastic. Lots might not arrive at the time assumed when starting a scheduler run. The production system needs to react dynamically to these situations, as well does the simulation. We have implemented different execution strategies:

- execute the schedule as-is, but perform only actions not violating any constraint,
- execute the schedule as-is, just logging the concerns,
- strictly adhere to schedule's lot IDs and process start dates, not allowing earlier process starts,
- consider only schedule's lot order.

Within our simulation model, we consider tool downs and sampling rates as stochastic events.

3.5 Speed up scheduler execution time

As the simulation model tends to come to a steady state only after a few simulation hours have passed, albeit started with a full fab model with current positions of material and tool states, we included an option to delay the first scheduler execution event by an arbitrary duration to avoid its time-consuming execution during the uninteresting warm-up phase.

In addition, we included an option to limit the maximum number of lots to supply to the scheduler (in that case, only the *i* highest ranked/nearest lots are kept) in case we ran into problems of the scheduler not finding a good solution if it considers too many lots.

For overall runtime of simulations, the ratio between *m* and *n* is essential. For many schedulers in real fabs, you will notice that it is close together because quality of the schedules depend on their calculation runtime. Within Section 5, we also run scenarios, where we change the ratio between *m* and *n* and evaluate the impact to the results due to the increasing uncertainty over time. The scheduler by itself is planning all lots and a schedule is available for much longer than *m*. Finally, this leads to the situation that a good assessment of a real fab scheduler can only be done in close to real time speed and even this does not guaranty optimality. Kim, Shim, Choi and Hwang (2003) present some techniques to overcome these issues and use a schedule for a longer time. However, we wanted to evaluate the scheduler as is and used several computers to run different simulations in parallel to get the results in Section 5 within a few weeks.

4 IMPLANT DISPATCHING ALGORITHMS

As a reference to compare the implant scheduler's decisions to, we decided to implement two different dispatching-only solutions, which will be presented briefly in the following.

4.1 Case-Differentiating Dispatching

The first dispatching algorithm is based on the following case differentiation:

- if any, choose highest-ranked allowed lot with current setup; else:
- if any, choose highest-ranked allowed lot with current species; else:
- if any, choose highest-ranked allowed lot; else:
- leave machine empty.

The algorithm could be considered a Simple Dispatching with Constraints and Same-Setup Rule (Duwayri, Mollaghasemi and Nazzal 2001). Despite its simplicity, it is expected to result in no constraint violations (as only allowed lots will be chosen) and should prevent setup/species changes to a limited extend.

However, as a typical dispatching algorithm, it handles each tool on its own and considers only the current material; for example, it cannot make the decision to wait for material coming in the desired setup/species within the next few seconds/minutes as a scheduler would probably do. Also it doesn't generally put lots onto the most suitable tool. An enhancement would begin dispatching at the least-flexible tool first to reduce the likelihood of tools having no suitable lots anymore, however even that cannot guarantee an optimal plan (Pinedo 2012).

Another problem of this approach is the inherent danger of starvation of lots with non-matching setup/species when lots of matching setup/species are consistently available. To prevent such situations, we introduced a configurable maximum allowed waiting time for setup/species changes, meaning that lots waiting for longer than this configured duration will be considered on the same level as lots requiring no setup/species change.

4.2 Weighted-Factors Dispatching

The second dispatching algorithm selects the lot having the smallest value calculated by the following cost function:

$$\begin{aligned} \mathit{cost} \quad := \quad & a * \mathit{rankingPosition} \\ + \quad & b * \mathit{if(setupChanged, cascadeFinishRemainingTime + setupTime)}\ [s] \\ + \quad & c * \mathit{isSpeciesChange}\ [1 \text{ or } 0] \\ + \quad & d * \mathit{isNotDesiredSpeciesChange}\ [1 \text{ or } 0] \\ + \quad & e * \mathit{remainingMinSpeciesTime}\ [s] \\ - \quad & f * \mathit{waitTime}\ [s] \end{aligned}$$

Obviously, the cost function can be parameterized by adapting the factors *a..f*. Also the cost components themselves could be interchanged freely. In comparison to the first algorithm, this one can include soft constraints like avoiding undesired species changes; also it does not suffer from the starvation problem because at some point, the waiting time will become so big that it overrules other terms and the lot will be considered more important than a lot of the current setup/species.

We implemented the described two approaches to have an additional baseline for comparing scheduler performance. Kim, Byung-Jun and So-Young (2010) describe additional approaches. However, we also compare the scheduler to a simple dispatch logic that ignores all implant constraints. This could be seen as a lower bound and gives us an indication about how good a scheduler could be at best; considering the fact that the costs of the optimal solution considering all constraints might be higher.

5 SIMULATION STUDY

The following important KPIs were compared between the Implant Scheduler, both presented dispatch algorithms and the one ignoring the special implant constraints altogether – as it is done in generic fab simulation.

5.1 Scenarios

Since we are using the scheduler in our fab, we do not have a dispatch rule in place that we can use for comparison. This is the reason, why we implemented the two approaches in Section 4. Table 3 shows the performed simulation runs to be compared. This allows us to assess them under a wide range of parameters

Table 3: Scenario parameterization.

Type	Parameters	# Scenarios
Scheduler, Upcoming material method 1, *m* =	{10min}, {20min}, {30min}	3
Scheduler, Upcoming material method 2, *m* =	{30min}	1
Case-D. Disp. max. WaitTime =	{inf}, {120h}, {48h}, {24h}	4
Weighted-F. Disp., Factors *a..f* =	{1, .1, 100, 100, .1, .01}, {1, .2, 100, 100, .1, .01}, {1, .3, 100, 100, .1, .01}, {1, .3, 100, 100, .1, .02}	4
Generic disp. (no IMP constraints)		1
		13 total

We use our current full fab simulation model with hundreds of different products and production flows, thousands of tools and start the simulation with a snapshot of the current material in the line so that we can reduce the warm-up phase to one week. After warm-up we simulate additional two weeks to capture the results. Because of the extensive runtime we did not make replications. Note, that running simulation for two weeks calls at least 700 times the scheduler. However, for a few scenarios that showed significant differences to others we confirmed the results with a second run. The KPIs in Table 4 were used for comparison of the different simulation scenarios.

5.2 Simulation results

Figure 3 shows that the scheduler clearly outperforms both of the simple dispatch algorithms regarding the number of setup changes performed. However, Figure 4 indicates that no significant difference could be found in the number of species changes that need to be done.

Table 4: Objectives/KPIs.

Objective	**Description**
Setup/Species changes	Number of changes to be minimized.
Throughput at Implanters	Wafer/h to be maximized.
Waiting time at Implanters	Value and standard deviation to be minimized.
Cycle Time	Value and standard deviation to be minimized.

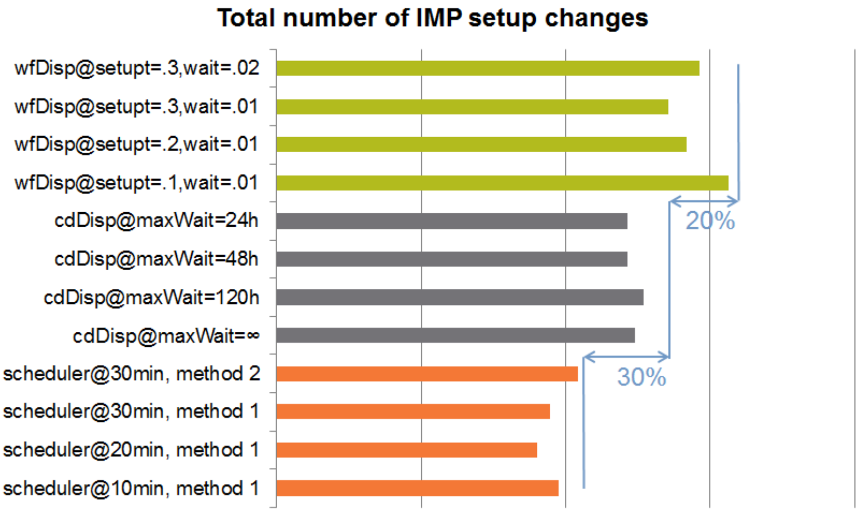

Figure 3: Total number of setup changes at implanters.

Another finding presented in Figure 5 is that the special implant constraints reduced the throughput of the implanters by about 3%, with all the methods whether dispatching algorithm or scheduler yielding similar results – with the exception of running the scheduler only once per 30 minutes in combination with upcoming material determination method 1. That indicates that this method does not deliver enough upcoming material for such a large *m*.

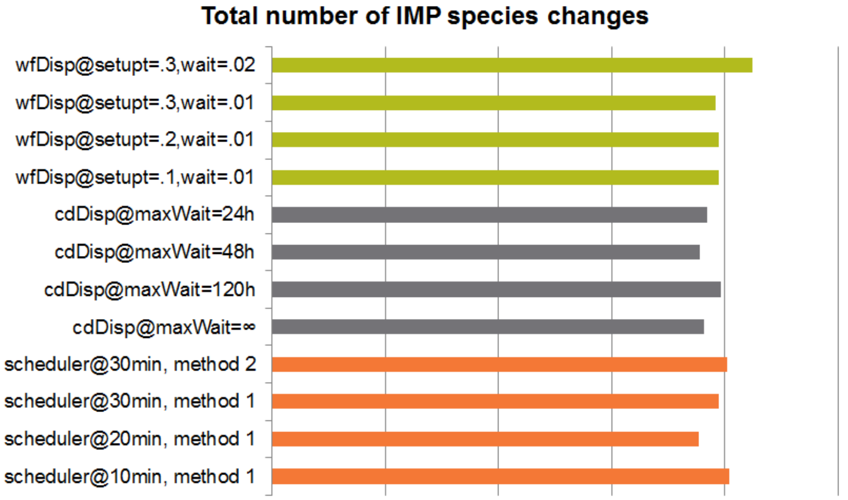

Figure 4: Total number of species changes at implanters.

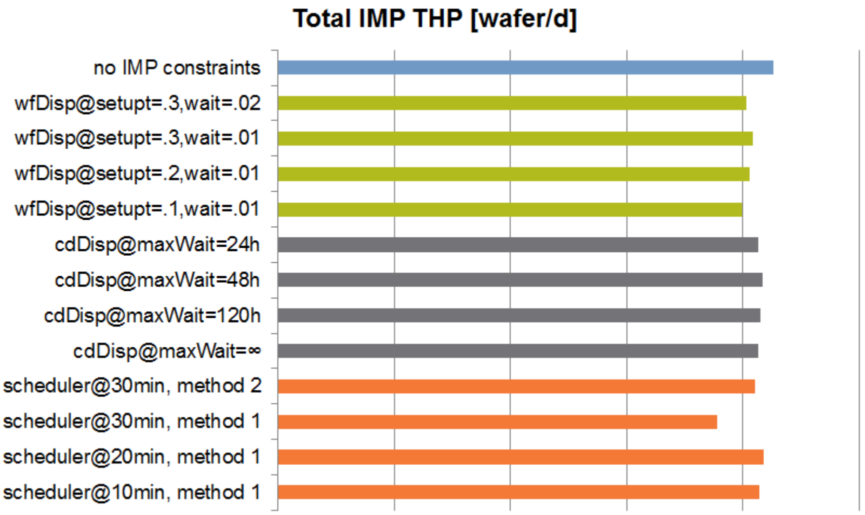

Figure 5: Total throughput at implanters [wafer/d].

By introducing the implant constraints into the simulation model, the waiting time at implant operations significantly increases, as can be seen in Figure 6. For the scheduler it can be seen that the lower we choose *m*, the more often it is called and the lower the waiting time will be. This is as expected. Also we see that while the maximum waiting time can be effectively controlled by the parameter of the case-differentiating dispatching: setting it to a low value has the drawback that the median waiting time increases.

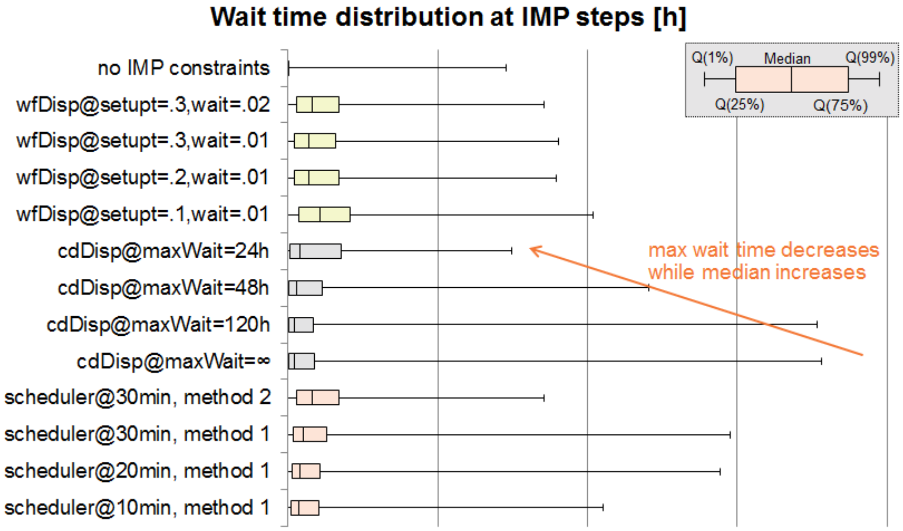

Figure 6: Waiting time distribution of lots at implanters [h].

Finally, a similarly large difference can also be seen in overall fab cycle time (Figure 7). Also, the scheduler is performing significantly better than the simple dispatch algorithms designed for comparison. Altogether we conclude that the results show that the scheduler outperforms the simple dispatching algorithms.

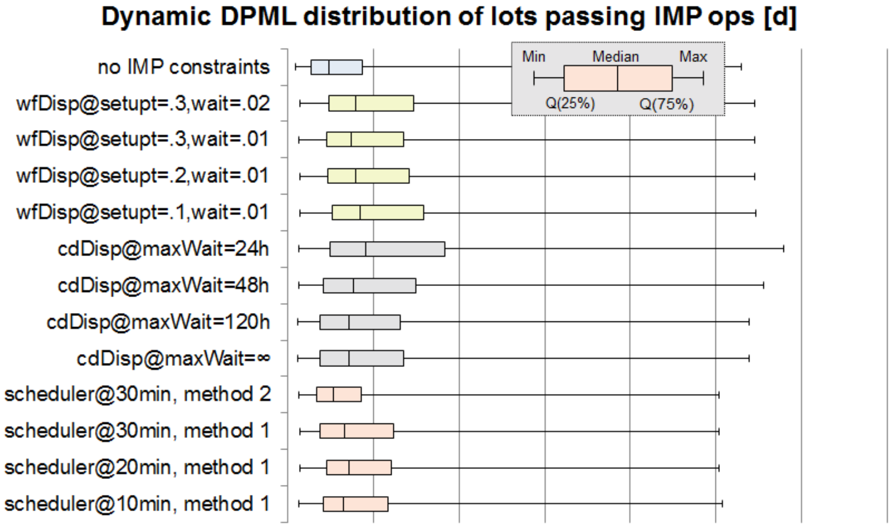

Figure 7: Overall fab Cycle Time (DPML) of lots passing at least one implant operation.

6 SUMMARY AND OUTLOOK

Simulation-based scheduler evaluation has been shown to be useful and practical for allowing the scheduler to be evaluated without risk and as is in a continuously running environment very similar to a real fab, enabling us not only to evaluate the performance of the scheduler itself, but also the influences on the entire fab's performance. The specialized dispatching algorithms can be used in day-to-day simulation runs as they yield more realistic results than generic dispatching (neglecting restrictions altogether) without incurring any visible runtime penalty.

As the scheduler was shown to be better than the simple dispatch rules, it might be interesting to implement a more sophisticated heuristic approach and comparing the scheduler to it. Duwayri, Mollaghasemi and Nazzal (2001) give some ideas in that direction. Another application – and more interesting for us – is to optimize the parameters of the scheduler to get better solutions and to understand the influence single parameters have on implanting and fab performance in different fab situations.

Besides that, the developed technique of linking a scheduler to the simulation can be re-used for evaluating any other scheduler just as well. This can be a big step towards risk mitigation before bringing new schedulers into production.

REFERENCES

Anders, A. 2000. *Handbook of Plasma Immersion Ion Implantation and Deposition*. Vol. 8. New York: Wiley.

Duwayri, Z., M. Mollaghasemi, and D. Nazzal. 2001. "Scheduling Setup Changes at Bottleneck Facilities in Semiconductor Manufacturing". In *Proceedings of the 2001 Winter Simulation Conference*, edited by M. Rohrer, D. Medeiros, B. A. Peters, and J. Smith, 1208–1214. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

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(2):290–298.

Kim, Y. D., J. Byung-Jun, and C. So-Young. 2010. "Scheduling Wafer Lots on Diffusion Machines in a Semiconductor Wafer Fabrication Facility". *IEEE Transactions on Semiconductor Manufacturing* 23(2):246–254.

Mönch, L., O. Rose, and R. Sturm. 2003. “A 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.

Mönch, L. 2007. “Simulation-Based Benchmarking of Production Control Schemes for Complex Manufacturing Systems“. *Control Engineering Practice* 15:1381–1393.

Pinedo, M. L. 2012. *Scheduling: Theory, Algorithms, and Systems*. New York: Springer.

Sprenger, R., and L. Mönch. 2012. “A Methodology to Solve Large-Scale Cooperative Transportation Planning Problems.” *European Journal of Operational Research* 23(3):626–636.

Winkler, T., P. Barthel, and R. Sprenger. 2016. “Modeling of Complex Decision Making using Forward Simulation”. In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. Huschka, S. Chick, J. Jimenez, P. Frazier, R. Szechtman, and E. Zhou, 2982–2991. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhang, P., Y. Lv, and J. Zhang. 2016. “An Improved Imperialist Competitive Algorithm Based Rolling Horizon Strategy for Photolithography Machines Scheduling“. *IFAC-PapersOnLine*, 49(12):1295–1300.

AUTHOR BIOGRAPHIES

THOMAS WINKLER is an engineer in the Line Analysis department (part of Industrial Engineering) of GLOBALFOUNDRIES Fab 1 in Dresden. He received a master’s degree in Applied Information Technologies at Dresden University of Applied Sciences. His research interests include capacity planning, heuristics for multiple-constraint problems, data visualization and simulation. His email address is thomas.winkler@globalfoundries.com.

RALF SPRENGER is manager of the Line Analysis department (part of Industrial Engineering) of GLOBALFOUNDRIES Fab 1 in Dresden. He received a Ph.D. from the Department of Mathematics and Computer Science at the University of Hagen, and a master’s degree in computer science at Dresden University of Technology. His research interests include industrial engineering in semiconductor manufacturing and optimization. His email address is ralf.sprenger@globalfoundries.com.