Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

ADVANCED SECONDARY RESOURCE CONTROL IN SEMICONDUCTOR LITHOGRAPHY AREAS: FROM THEORY TO PRACTICE

Dirk Doleschal
Gerald Weigert

Electronics Packaging Laboratory
Technische Universität Dresden
Helmholtzstraße 18
01062 Dresden, GERMANY

Andreas Klemmt
Frank Lehmann

Infineon Technologies
Königsbrücker Straße 180

01099 Dresden, GERMANY

ABSTRACT

Semiconductor frontend fabs are very complex manufacturing systems. Typically, the bottleneck of such a fab is the photolithography area because of its highly expensive equipment and the huge number of required secondary resources – the so called reticles. A reticle (mask) is needed to structure different layers of integrated circuits on the wafers. The reticles can be moved between the equipment with regard to several constraints. This paper examines the benefits of a solver-based reticle allocation in comparison to a classical rule-based heuristic. In a first part, several simulation experiments are performed on the basis of representative test data. The second part presents results from real world application. Thereby it is shown, that the new approach shows significant improvements of different key performance indicators (KPIs).

1 INTRODUCTION

Most of the practice-oriented scheduling tasks are NP-hard optimization problems (Brucker 2004). Hence, for solving complex scheduling problems, a lot of heuristics and decomposition methods were developed and investigated. A comprehensive overview of several of such approaches can be found for instance in Gupta and Sivakumar (2002) or Ovacik and Uzsoy (1997). Thereby, problem-specific heuristics in combination with simulation and scheduling systems have shown the best efficiency.

In this paper the lithography area of a high mix – low volume semiconductor fabrication is investigated. Usually the photolithography area is a bottleneck work center of a wafer fab because of its highly expensive machines and its complex process constraints (cf. Chung and Huang 2008). So, an effective planning of the photolithography area will have a high practical relevance for the whole fab. Generally, in this process a resist is structured to act as a direct mask for subsequent structuring of the underlying substrate material. The photolithography process comprises several sub-processes. Firstly, adhesives are added and moisture is removed from the surface. This is followed by a resist coating, the exposure process and the development of the resist. Finally, there is a curing and an inspection of the resist. The main photolithography process is the exposure. Here a reticle (mask) is used to structure a resist layer with the desired circuit pattern. So, for every new layer with a changing pattern, the reticle has to be changed. Since integrated circuits are commonly created layer by layer, many cycles of these photolithography processes are performed. Taking into account that channel widths are more and more shrinking, the layer-to-layer alignment, called vertical dedication in the following, is increasingly important for achieving a respectable yield. Even equal lenses are individually slightly different in their characteristics. For this, some wafer always has to be processed on the same lithography unit. Due to the fact, that the reticle is really expensive, mostly only one reticle for each type of structure exist. Also the process time can vary between two

allowed machines, because some machines are older and some are newer and the process speed is increased for newer generations of lithography machines. Furthermore, additional costs can occur, if a send ahead lot or wafer is necessary. More detailed information about such a lithography area can be found in Mönch et al. 2001.

One of the keys for reaching good KPIs in the lithography area is an effective management of reticle- and lot-to-tool allocations with regard to the presented process constraints. For this, a new approach based on iteratively solved mixed integer programs is presented in the following.

The paper is structured as follows. In section 2 the problem is described and the simulation model and the objectives are presented. Section 3 is used to define the mixed integer programming model. In section 4 the test setup is presented and afterwards in section 5 the results are shown. A short information about the practical usage is given in section 6 and a short conclusion is done in section 7.

2 PROBLEM DESCRIPTION

The underlying scheduling problem is a single operation problem with unrelated parallel machines, release dates, setups, dedications and secondary resources. Such problems can often be found in cases where not only a machine but also additional equipment (secondary resources) is needed for production. In the presented case, the secondary resources are the reticles. For the rest of the paper it is assumed that: exactly one secondary resource is needed for processing a job (lot); each secondary resource exists only one time; and different jobs can require the same secondary resource. So, secondary resources can also be defined as product families. In the $\alpha \mid \beta \mid \gamma$ notation of Graham et al. (1979), extended by Pinedo 2008 and Baptiste et al. 2001, this problem can be written as $R_m \mid r_{ij}, s, p_{ij}, aux, M_i \mid O$, where O is one of the objectives, defined in the following sections.

2.1 Problem Parameter

The scheduling problem consists of the following elements (with regard to the specificities of the lithography area):

- n different secondary resources F_i (i=1,…,n), where every secondary resource/family includes n_i jobs,
- m different parallel machines M_k (k = 1,…,m),
- A dedication matrix $D \in \{0,1\}^{n \times m}$, which specifies permitted and disabled machines for each product. Also $D_k := \{ i \mid D_{ik} = 1 \}$ is the set of products permitted for processing on machine M_k. In the same manner, $D_i := \{k \mid D_{ik} = 1\}$ is the set of machines permitted for processing family F_i,
- $p_{ik} > 0$ is the processing time for a job of product family F_i on machine M_k if $D_{ik} = 1$,
- Each job j has a release date r_j and a due date d_j,
- $s > 0$ is the setup time, which occurs, if on a machine the processing with the secondary resource changes,
- Each job j has a priority j_p. This priority is important for scheduling,
- A job j of the family F_i can be vertical dedicated. In this case, the job can be only processed on a special machine out of the set D_i,
- For all combinations of secondary resource F_i and permitted machines D_i a time dependent cost function c^R is defined, which set the cost for production of a product from type F_i on a machine from the set D_i. These cost occur if a family needs a send ahead lot, for example.

Now the challenge is to generate a valid schedule with the primary goal of a good load balancing on the machines and to hold the operational due date. The creation of a schedule is done by a DES-system (discrete event simulation system) with a practicable dispatching rule.

2.2 Adapted Simulation Model with Simplifications

A simulation model is built up to generate schedules for this problem class. The used DES-system is the simcron MODELLER. Here the needed constraints for example for the secondary resources are implemented. The important processing constraints are:

- Secondary resource constraints:
  - A secondary resource has to be assigned to the machine, where the associated jobs are to be processed,
  - Only one secondary resource for each type exists, so only one machine can be equipped with this resource at a time,
  - A secondary resource can be stored in a central storage system or in a machine.
- Machine constraints:
  - Each machine can hold a predefined number of reticles at an instant of time, whereby only one of these secondary resources is used for processing,
  - The job capacity of a machine is one, so only one job can be processed at a time.
- Setup constraints:
  - A setup time *s* occurs, if the setup state of a machine has to be changed to another secondary resource. This setup time *s* depends on where the new secondary resource is located. If the new used secondary resource is hold in the same machine, than the setup time *s* is small, otherwise the setup time *s* is higher, because the secondary resource has to be transported to the machine.
- Processing constraints:
  - The processing time depends also on the cost function c^R. This cost function has an effect on the real processing time and an additional delay after finishing on a lithography machine. For example, if a send ahead lot is necessary, the processing time doubles and an additional delay of five hours is inserted in which the lot is measured.

2.3 Objectives

To measure the performance of the tested scheduling approaches, the following objectives are used:

- Tardiness $T_j = \max(0, c_j - d_j)$
- Lateness $L_j = c_j - d_j$
- Cycle time $C_j = c_j - r_j$
- Flow factor $FF_j = p_j / C_j$
- Equipment utilization U_k
- Number of setups / reticle moves

Thereby, c_j is the completion time, d_j is the (operational) due date, p_j is the process time and r_j is the release time from job *j*. Also, the operating curve is analyzed. The operating curve represents the relationship between the utilization and the flow factor. To fit the curve, the following function is used, where *x* is the utilization and $f(x)$ is the corresponding flow factor:

$$f(x) = \alpha \frac{x}{1-x} + 1 \qquad 0 \le x < 1 .$$

Here, α is defined as a variability coefficient. The lower α is, the better is the work center balanced. To fit such an operating curve, for each logical work center, tuples are generated. For this the simulated time is divided into time slots (e.g. 1 day) and then for all machines the average utilization and for all lots

which are processed within this time slot on these machines the average flow factor is calculated. For these points the α which fits the points best is calculated and the curves are drawn.

2.4 Used Dispatching Rule

DES systems mostly work with dispatching rules like FIFO/SPT or due date related rules like EDD (earliest due date) or ODD (operational due date, cf. Rose 2003). Within this paper a practicable dispatching rule was generated to get good schedules regarding to the priority, cost function and number of setups. In this section, this dispatching rule – called PrioODD – is described in a short way. This is done from the viewpoint of the scheduler within the used DES system. This scheduler tries to put jobs from a queue to an allowed machine. Thereby, the first job in the queue is used first and after this the other jobs are tried in their order. Each job j gets a priority value j_{prio}, dependent on the job priority j_p and the operational due date d_j. This priority value is build up in the following way:

$$j_{\text{prio}} = j_p \cdot K - d_j \qquad \text{where } K > \max_j(d_j)$$

With this construct it is ensured that the jobs j are firstly sorted with their priority j_p and secondly with increasingly operational due date d_j. All jobs are sorted in the queue in the order of their priority value, so a priority and operational due date scheduling is ensured. Now, the scheduler tries to put the first job j (of family F_i) from the queue (job with highest priority j_{prio}) to the machine m. Next to the priority calculation, the rule logic needs to be explained. This is done in Figure 1.

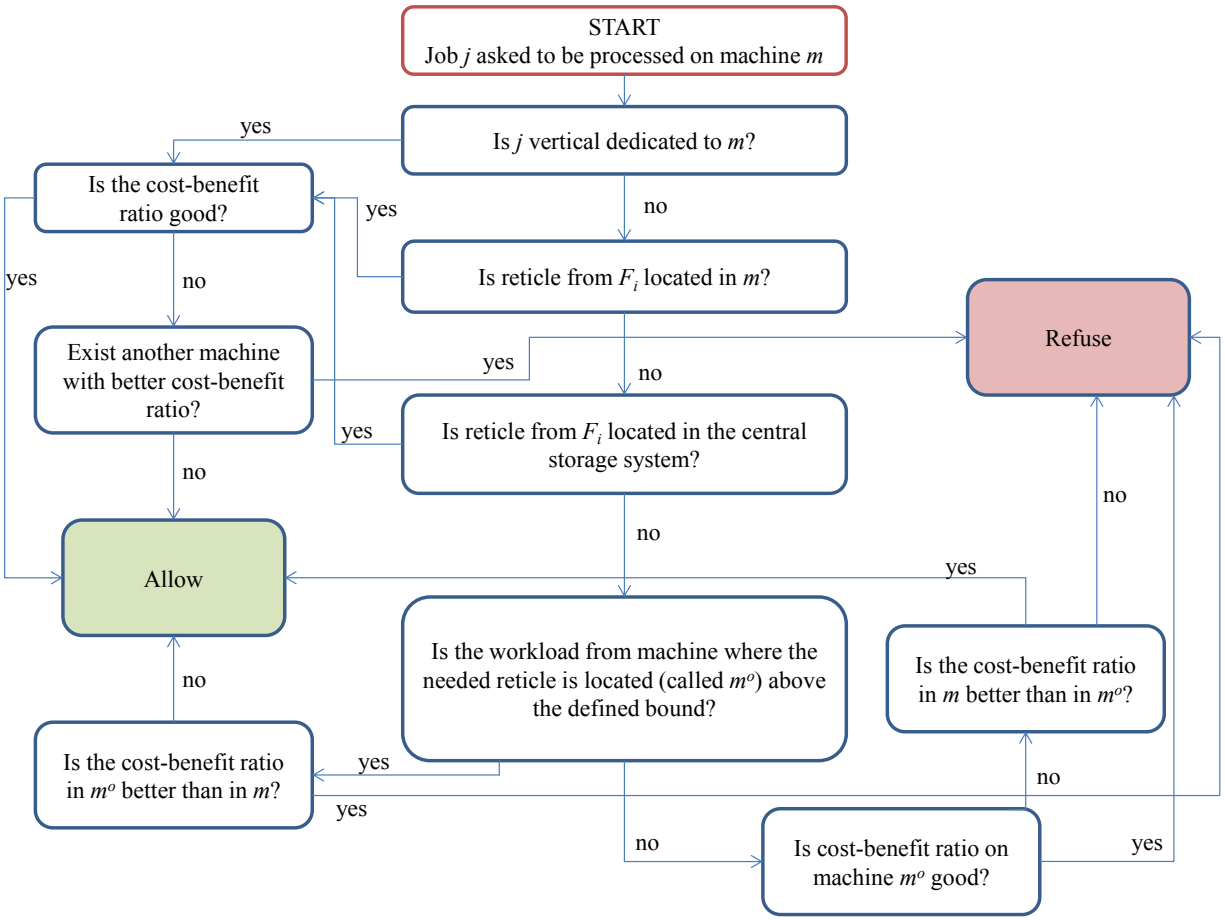

Figure 1: Used dispatching rule in the simulation model

Here a workload bound is defined, which is used in the way, that a machine can submit a reticle to another machine, if its workload is too high. With this bound a manual intervention as it occurs in a real environment is simulated.

The complexity of the flow chart already shows the complexity of the investigated reference rule. If a job j of family F_i is refused within this rule, the next allowed machine for this family F_i is tested with the same rule. Only if all machines have refused this job, the next job j (regarding priority j_{prio}) is tested.

This rule primarily aims on the minimization of setups – whereby the cost function is observed. The ODD part in the sorting algorithm observes tardiness and lateness. Even if this (relatively) intuitive rule, significantly differs from the much more complex fab rule it is still a good reference for rule based decision making.

3 MIXED INTEGER PROGRAMMING MODEL

In this section, the simulation model using the dispatching rule defined in section 2.4 is improved by an additional capacity allocation step, based on mixed integer programming (MIP). For this step only "static" information are necessary. The capacity allocation step is called secondary resource planner (SRP) in the following. It statically assigns the reticles to the equipment. To get an dynamic schedule, the SRP is called cyclic from the DES model and the received information (which reticle has to be assigned to which machine) is used in the simulation model until the next process call. The underlying MIP model for the SRP is first published in Klemmt et al. 2010. A more detailed description is given in Klemmt 2012. Nevertheless, in this paper the used MIP model is described in detail in the following section. Thereby, some changes are done, to fit the problem defined in section 2.

3.1 Required Information

The needed information for the SRP can be divided into static and dynamic information. The static information is not time dependent; this means it is independent from the current scheduling state. In contrast to this, the dynamic information changes in dependence of the scheduling state. The following parameters are used in the SRP:

Static parameters:

D_k	Set of all secondary resources, allowed for machine $k \in M$
p_{vk}	Processing time for one job of secondary resource $v \in D_k$ on machine $k \in M$

Dynamic parameters:

n_{vk}	Number of vertical dedicated jobs for secondary resource $v \in D_k$ on machine $k \in M$
l_{vk}	Secondary resource $v \in D_k$ is located on machine $k \in M$, than $l_{vk} = 1$, else $l_{vk} = 0$
u_k	Remaining processing time for machine $k \in M$
w_v	Cumulated number of jobs for secondary resource v
c_v^J	Cumulated job priority for secondary resource v
c_{vk}^J	Cumulated job priority for secondary resource $v \in D_k$ on machine $k \in M$ (where $n_{vk} > 0$)
c_{vk}^{SR}	Cost to allocate secondary resource $v \in D_k$ to machine $k \in M$

Furthermore, the values c_v^J and c_{vk}^J are used to calculate the value $c_{vk}^{\max}$ which defines the maximum job priorities of secondary resource v which can be processed on machine k, where the vertical dedication is considered:

$$c_{vk}^{\max} = c_v^J - \sum_{l \in M(l \neq k, v \in D_l)} c_{vk}^J .$$

3.2 Variables

To define a mixed integer capacity planning algorithm, unknown variables have to be defined:

$Q_{vk} \in \mathbb{Z}$	Number of jobs from secondary resource $v \in D_k$, which should be processed on machine $k \in M$,
$D_{vk} \in \{0,1\}$	Secondary resource $v \in D_k$ is assigned to machine $k \in M$, 0 otherwise,
$I_k \in \{0,1\}$	Machine $k \in M$ is assigned at least one secondary resource, 0 otherwise,
$B^U \in \mathbb{Z}$	Upper bound for the maximum workload over all machines.

With the defined parameters and the unknowns the mixed integer based capacity planning model can be formulated. Thereby, this model is divided into three stages, where each stage uses the result from the previous stage.

3.3 Stage 1

The first optimization objective is to find the reticle – machine combinations which have the highest positive impact on the cost-benefit ratio. To achieve this goal the first optimization model is constructed in the following way:

Optimization model 1

$$\sum_{k \in M} \sum_{v \in D_k} (c_{vk}^{SR} - c_{vk}^{\max}) D_{vk} \rightarrow \min \qquad \text{subject to} \tag{1}$$

$$Q_{vk} + \sum_{l \in M(l \neq k, v \in D_l)} n_{vl} \leq w_v \qquad k \in M, v \in D_k \tag{2}$$

$$Q_{vk} \leq D_{vk} w_v \qquad k \in M, v \in D_k \tag{3}$$

$$D_{vk} \leq Q_{vk} \qquad k \in M, v \in D_k \tag{4}$$

$$\sum_{k \in M(v \in D_k)} D_{vk} \leq 1 \qquad v \in F \tag{5}$$

The objective function (1) has the goal to optimize the work in progress in this way that as many high priority jobs are allocated with as low cost as possible. With equation (2) the maximum number of jobs which can be allocated to a secondary resource – machine combination is set. Constraint (3) forces that D_{vk} has to be 1 if this allocation is used. Vice versa equation (4) assures that jobs are allocated if $D_{vk} = 1$. The last constraint (5) ensures that each secondary resource is only used once. Now optimization model 1 has to be solved and D_{vk}^* is an optimal solution. Than the value z_C^* is the best objective value, defined as:

$$z_C^* := \sum_{k \in M} \sum_{v \in D_k} (c_{vk}{}^R - c_{vk}{}^{\max}) D_{vk}^* .$$

With this solution, the set of all secondary resources SR^* where the priority value is higher than the cost on at least one machine can be defined as follows:

$$SR^* := \{v \in F \mid \exists D_{vk}^* = 1, k \in M, v \in D_k\},$$

where z_C^* and SR^* are now used in the second stage of the capacity planning model.

3.4 Stage 2

This stage is used to maximize the number of allocated jobs.

Optimization model 2

$$\sum_{k \in M} \sum_{v \in D_k} Q_{vk} \rightarrow \max \qquad \text{subject to} \tag{6}$$

$$\sum_{k \in M} \sum_{v \in D_k \cap SR^*} (c_{vk}^R - c_{vk}^{\max}) D_{vk} = z_C^*, \tag{7}$$

and (2) – (5).

The objective function (6) is used to reach the goal of maximizing the number of allocated jobs. Thereby, equation (7) ensures that the optimized allocation cost z_C^* from stage 1 is held for all secondary resources of the set SR^*. If Q_{vk}^* is an optimal solution for this second stage, the objective value z_P^* is defined as:

$$z_P^* := \sum_{k \in M} \sum_{v \in D_k} Q_{vk}^*,$$

where z_P^* and the results from the first stage is used in the last stage as input parameters.

3.5 Stage 3

The optimization model in this stage has the objective to balance the load over all machines and in the same way, to ensure, that all machines have a reticle and the number of reticle moves is reduced within this stage.

Optimization model 3

$$B^U - \sum_{k \in M} I_k + \sum_{k \in M} \sum_{v \in D_k} ((c_{vk}^{SR} - c_{vk}^{\max}) K_1 - l_{vk} \cdot K_2) D_{vk} \rightarrow \min \qquad \text{subject to} \tag{8}$$

$$\sum_{v \in D_k} D_{vk} \le n I_k \qquad k \in M \tag{9}$$

$$\sum_{v \in D_k} D_{vk} \ge I_k \qquad k \in M \tag{10}$$

$$\sum_{k \in M} \sum_{v \in D_k} Q_{vk}^* = z_P^* \tag{11}$$

$$\sum_{v \in D_k} Q_{vk} p_{vk} + u_k \le B^U \qquad k \in M \tag{12}$$

and (2) – (5) and (7).

The multi-criteria objective function (8) is used to ensure that the maximal workload B^U over all machines is minimized. Concurrently the number of used machines I_k has to be maximized. Also the number of reticle moves should be reduced with the last summand while the cost are also considered. Thereby the values K_1 and K_2 are scaling factors. The maximum workload is limited by constraint (12). Equation (11) has the effect that the result from the second stage is held. The parameter I_k is limited by the constraints (9) and (10).

The result from this third stage of the secondary resource planner is an allocation D_{vk}, which assigns secondary resources to machines. The allocation is further used in the described simulation model.

4 TEST SETUP

This section describes how the secondary resource allocation model is integrated into the simulation model presented in the second section. Furthermore, the test instances are described in a short way.

4.1 Implementation

The secondary resource planner is called cyclic within the simulation model. This means, every X minutes in the simulation time the described mixed integer model is called and the result from this model is used within the simulation model (see Figure 2, $X > 0$). Thereby, the capacity planning algorithm can get forecast information for jobs which arrive at the work center within the next time.

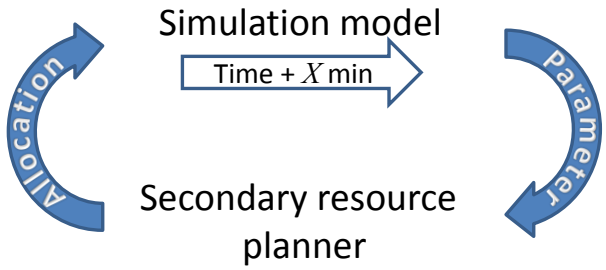

Figure 2: Implementation of secondary resource planner

Because of the secondary resource allocation through the MIP model, the dispatching rule utilized in the simulation model can be designed easier. From the viewpoint of the scheduler, each job which asks for processing is allowed on a machine. This is because the result from the capacity planning model is interpreted in the way that only assigned machines are allowed for a secondary resource. Non-assigned machines are disabled for a secondary resource family. So, this dispatching rule is much simpler than the rule described in section 2.5.

4.2 Test Instances

To test the described method, practice orientated test instances or benchmarks are generated. As a role model for this, a high-mix low-volume semiconductor industry area was used. As described in the beginning, the lithography area, where reticles are the secondary resources, is used. The test instances are generated similar to the test instances in Doleschal et al. 2013. To get more practice orientated test instances, the number of reticles and jobs as well as the release dates are directly retrieved from the cooperating industry partner. For this, the data from about six weeks is retrieved and used within the benchmark. The dedication matrix is generated randomly, with regard to the degrees of freedom existing in the underlying manufacturing system. Also, the process times are generated this way, assuming that the theoretical total utilization of the machines are 60%, 70% or 80%. This utilization is just a lower bound, because no setup times and additional process times due to the cost function is considered. The cost function is generated

randomly. Also the process times can be homogenously or heterogeneously distributed over the allowed machines. In summary, 30 test instances are generated, where each combination of utilization and homogenous or heterogeneous process time is generated 5 times. The following methods are tested:

- MIP_{20}^{0}, MIP_{20}^{60}, MIP_{20}^{120}, MIP_{60}^{120},
- $PrioODD_{20}$, $PrioODD_{50}$, $PrioODD_{100}$.

Thereby the MIP rules are the dispatching rules using the SRP where the lower value describes the time interval between two SRP runs. The upper value describes the forecast horizon (look ahead). The lower value for PrioODD defines the workload bound.

5 RESULTS

Now the results for the used methods and benchmarks are presented. Here the average tardiness, cycle time, lateness and flow factor are calculated for all lots. The utilization and the standard deviation for the utilization are calculated for all machines in a defined time interval, where the processing is stable (for the 6 weeks in the benchmark, week 2 – 5 is used). The average number of setups is calculated for the whole simulation time. All bar figures are normalized to the result gained by the dispatching rule $PrioODD_{100}$. Figure 3 shows the results for all test instances.

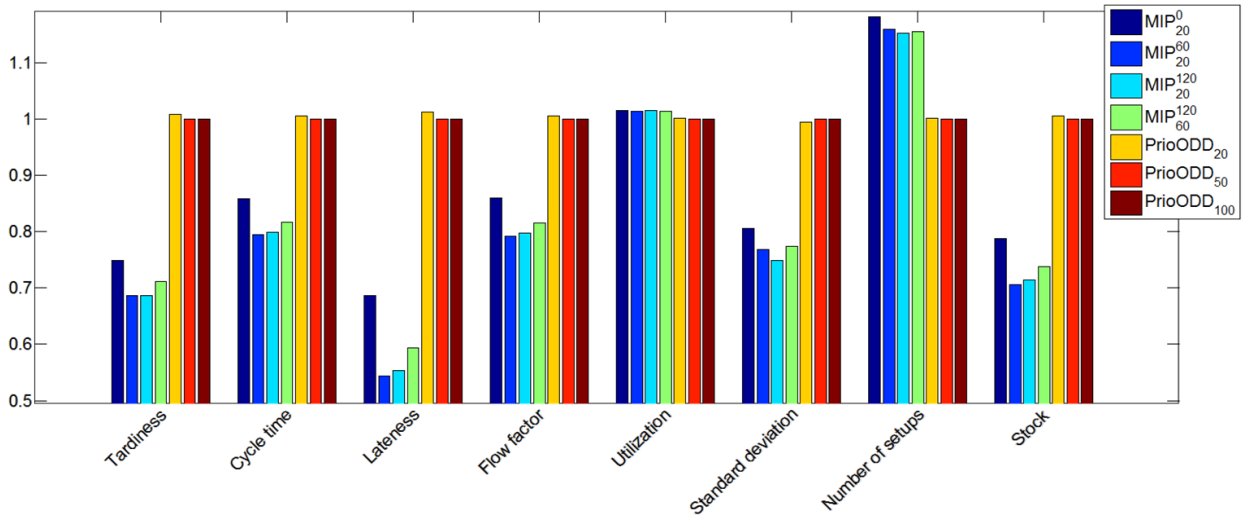

Figure 3: Normalized results for all scheduling methods and all test instances

This graphic shows that the mixed integer method performs significantly better. All time and lot based objectives like tardiness, cycle time, lateness and flow factor could be reduce. Only the number of setups increases slightly. The SRP approach performs better if it is called more often and a forecast horizon is considered. Also, the utilization is increased slightly whereby the standard deviation is decreased – so the equipment are more balanced. The average lot stock correlates the same way. According to the definition of lateness, the lateness could also get negative values. In Figure 3 all lateness results are positive, which means, over all test instances the jobs are too late according to their due date in average for all methods. For test instances with a low total utilization the values for lateness could also be negative.

But not only the average values for the lateness is important, also the spread of the lateness values for each lot has to be regarded. So, in Figure 4 the distribution of the lateness for one example (utilization = 60%) is shown. That means, it shows how good the lots meat the operational due date (marked as 0). Here the MIP methods also have a better distribution compared to the dispatching rule.

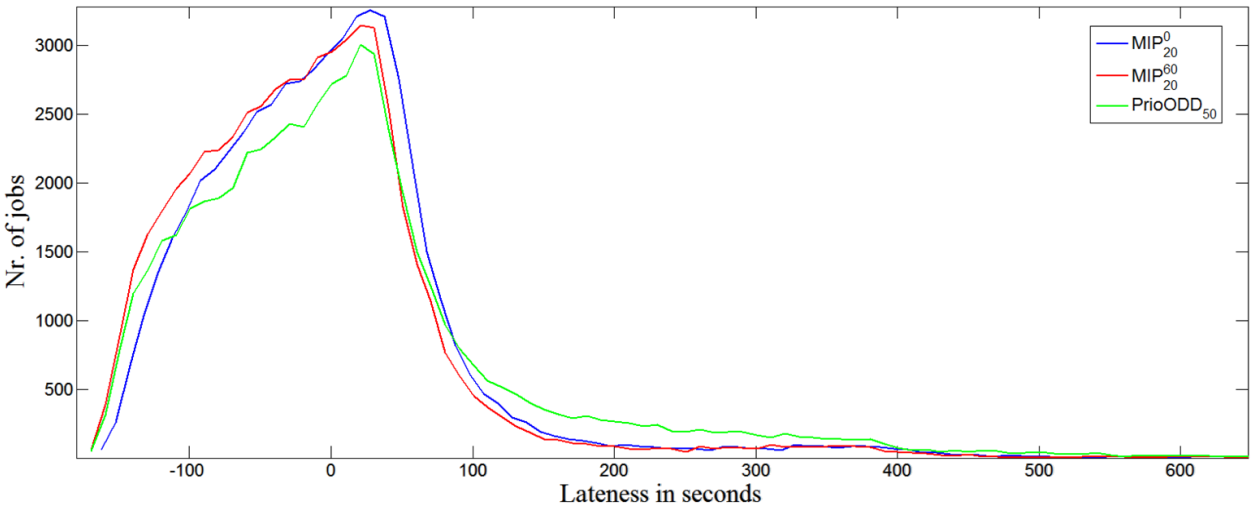

Figure 4: Distribution of the lateness for one benchmark (utilization of 60%, homogenous process times)

As described in section 2.3 operating curves are used to represent the variability in a work center. Such an operating curve is shown Figure 5. The used test instance has an theoretical utilization of 80% and heterogeneous process times. The x-axis shows the average utilization and the y-axis the average flow factor.

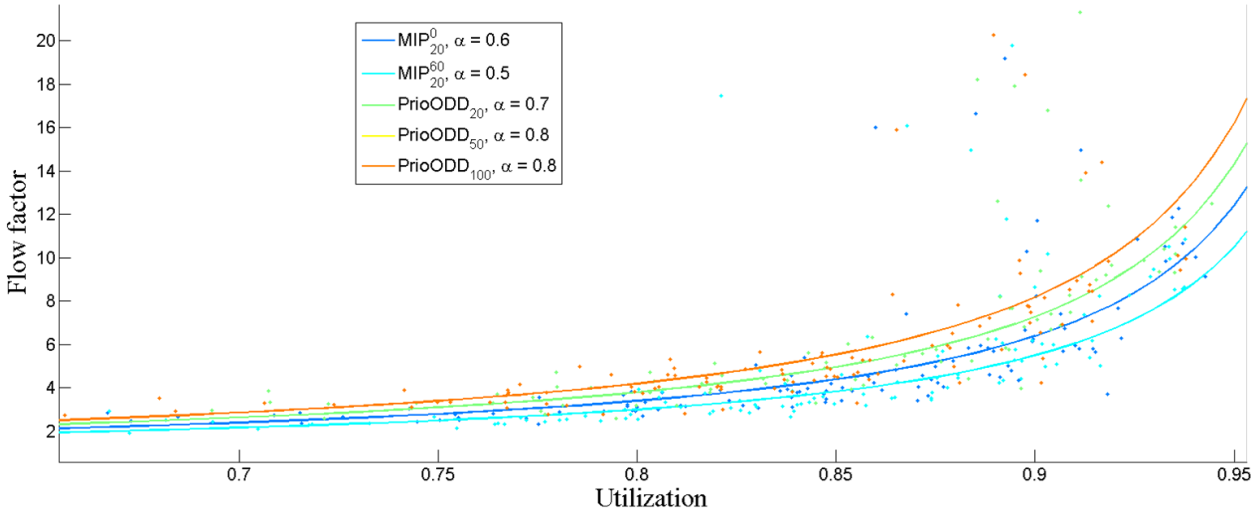

Figure 5: Operating curves for a test instance with 80% utilization and heterogeneous process times.

Overall the described mixed integer based capacity algorithm gains very good results with this practical orientated test instances. Also the used dispatching rule PrioODD generates practicable schedules, whereby the dispatching rule needs fewer reticle moves.

6 PRACTICAL EXPERIENCE

Even if the presented approach above, differs from implementation in the fab, it is still a good reference. It helps in understanding the system performance (coupling static and dynamic methods), in parameterizing variables (rescheduling intervals, look ahead, queue length parameters etc.) and in evaluating optimization potentials (estimated throughput, cycle time, lateness effects).

Implementing this approach in the real fab comes along with a lot of additional challenges. On the one hand the model has to be extended for handling some constraints which are not discussed in the paper (e.g. existence of reticles copies; much more complex cost function; transportation policies; inclusion of previous decisions). On the other hand the integration of such an approach into fab logic should not be underestimated (online replicated repositories for dispatching system, rule backups etc.). But even if the "installation effort" is high, it is legitimated by the reached results. After turning on the advanced SRP all KPIs nearly show the behavior as estimated in Figure 5.

7 CONCLUSION

In the past simulation and mathematical methods were often competing offers for planning and control of manufacturing processes. But it has shown that both traditional methods – discrete event simulation as well as mathematical programming (i.e. mixed integer programming) – as a single application are not suitable to solve more complex problems. Simulation typically works rules based and time directed which is limiting optimization options. Dynamic mathematical programming formulations often fail due to complexity. But the combination of both approaches has the ability to solve complex practical problems. The principle is the alternate use of simulation and mathematical optimization to use the advantages of both methods (static exact optimization and dynamic simulation). On the example of the reticle allocation management in the lithography area it is shown, that an online application of such an approach is possible for the operational planning and control. Even if the implementation effort is high, the runtime effort is reduced drastically (e.g. optimization time). Benchmark investigation as well as practical experiences show, that the combined approach significantly outperforms classical rule based dispatch approaches for nearly all considered objectives.

ACKNOWLEDGMENTS

This work was supported by the Federal Ministry of Education and Research of Germany (promotion number 16N11588).

REFERENCES

Baptiste, P., C. Le Pape and W. Nuijten, 2001. *Constraint-based scheduling: applying constraint programming to scheduling problems*. Kluwer Academic Publishers.

Brucker, P. 2004. *Scheduling algorithms*. Springer.

Chung, S.H., C.Y. Huang, and A.H.I. Lee. 2008. *Heuristic algorithms to solve the capacity allocation problem in photolithography area (CAPPA)*. In *OR Spectrum* 30:431-452.

Doleschal D., J. Lange and G. Weigert. 2013. "A simulation study on a mixed integer based scheduler for secondary resources in a parallel machine work center problem based on a high mix – low volume production." In *22nd International Conference on Production Research*.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. 1979. "Optimization and approximation in deterministic sequencing and scheduling: a survey." In *Annals of Discrete Mathematics*, Vol.5, 287-326.

Gupta, A. K. and A. I. Sivakumar. 2002. "Simulation based multiobjective schedule optimization in semiconductor manufacturing." In *Proceedings of the 2002 Winter Simulation Conference*, 1862-1870.

Klemmt, A., J. Lange, G. Weigert, F. Lehmann and J. Seyfert. 2010. "A multistage mathematical programming based scheduling approach for the photolitography area in semiconductor manufacturing." In *Proceedings of the 2010 Winter Simulation Conference*, 2474–2485.

Klemmt A. 2012. *Ablaufplanung in der Halbleiter- und Elektronikproduktion: hybride Optimierungsverfahren und Dekompositionstechniken*. Vieweg + Teubner.

Mönch, L., M. Prause, and V. Schmalfuss. 2001. "QSimulation-based solution of load-balancing problems in the photolithography area of a semiconductor wafer fabrication facility." In *Proceedings of the 33nd conference on Winter simulation*, 1170-1177.

Ovacik, I. M., and R. Uzsoy. 1997. *Decomposition methods for complex factory scheduling problems*. Kluwer Academic Publishers.

Pinedo, M. 2008. *Scheduling: theory, algorithms and systems*. Springer.

Rose, O. 2003. "Accelerating products under due-date oriented dispatching rules in semiconductor manufacturing." In *Proceedings of the 2003 Winter Simulation Conference*, 1346-1350.

AUTHOR BIOGRAPHIES

DIRK DOLESCHAL studied mathematics at Dresden University of Technology, Germany. He obtained his degree in 2010 in the field of optimization. He has been a Research Assistant at Electronics Packaging Laboratory of the Dresden University of Technology since 2010 and works on the field of production control, simulation & optimization of manufacturing processes. His email is Dirk.Doleschal@tu-dresden.de.

GERALD WEIGERT is an Assistant Professor at Electronics Packaging Laboratory of the Dresden University of Technology. Dr. Weigert works on the field of production control, simulation & optimization of manufacturing processes, especially in electronics and semiconductor industry. He was involved in development of simulation systems as well as in its application in industrial projects for scheduling. His email is Gerald.Weigert@tu-dresden.de.

ANDREAS KLEMMT received his master's degree in mathematics in 2005 and Ph.D. in 2011 at the Dresden University of Technology. He is employed as staff engineer in the operations research and engineering group of Infineon. His current research interests are capacity planning, production control, simulation & optimization. His email is Andreas.Klemmt@infineon.com.

FRANK LEHMANN received his master's degree in Electrical Engineering at the Dresden University of Technology. He works as a senior staff engineer within the Factory Logistics and Automation group of Infineon Dresden and is responsible for the RTD team and WIP flow management improvement projects. His email address is Frank.Lehmann@infineon.com.