Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

ONE SOLVER FOR ALL - A GENERIC ALLOCATION CONCEPT FOR PLANNING AND SHOP FLOOR CONTROL

Sebastian Werner
Frank Lehmann
Andreas Klemmt
Joerg Domaschke

Infineon Technologies Dresden
Königsbrücker Str. 180
01099 Dresden, Germany

ABSTRACT

This paper gives an overview of several optimization solutions for semiconductor problems using mixed integer programming (MIP). The single solutions presented in former papers are not key of the publication. We rather focus on the generic portion within each solution and the approach of building a unique MIP model. This allows us to reduce complexity in different applications. The universal model enables the use in a wide range of problems for different optimization stages mapped to static allocation problems. The model itself is a kit of constraints that can be activated according to the problem needs. The underlying data layer is an abstract database model that can be fed by different data sources. The paper describes the advantages of the consistent technical embedding of database, different solvers and generic MIP models in the MES environment.

1 INTRODUCTION

Algorithm based systems have been applied to semiconductor shop floor control systems for many years. Graphic modeling interfaces as e.g. provided by Real Time Dispatcher™ (RTD) have become a standard in the industry and allowed rules to be designed by advanced technicians which did not need to be IT specialists anymore. Thus production experts combining experiences in technology, logistics and basic optimization techniques could implement not only simple sequencing rules but even fairly complex approaches within reasonable time horizons and fast adaptation cycles if required.

However, while all of these algorithm based solutions helped to improve logistics performance they still remain fixed procedures which do not necessarily provide optimal solution. There is still a gap of productivity to utilize which may become remarkable if adequate monitoring is not in place. This needs to be established and maintained additionally, as well as all the manual rule adaptations whenever basic constraints change. Furthermore, system complexity increases with all of the solutions mentioned above and some of the algorithms – e.g. loop calculations – may push standard dispatching systems to their limits in terms of modeling capability and response times.

Meanwhile, the mathematical programming as an alternative optimization technique plays an important role in the semiconductor industry (Mönch et al. 2011; Klemmt 2012; Bixby, Burda, and Miller 2006). It provides target function or constraint based self adapting systems. Highly customized but expensive commercial scheduling solutions are available.

A significant part of corresponding productivity potentials could be addressed by models solving static allocation problems (cf. Akcali, Üngör, and Uzsoy 2005; Toktay, and Uzsoy 1998; Chung, Huang, and Lee 2006; Chung, Huang, and Lee 2008; Doleschal, Lange, and Weigert 2012). This approach provides

results within shortest calculation time and covers a broad range of optimization problems arising in semiconductor industry. However, expert resources providing knowledge to generate and maintain mathematical models, in many variants, are rather limited. Therefore, the variety of different models needs to be minimized by standardization.

The paper is organized as follows: In section 2 we give a motivation for a generic allocation concept. Therefore, the related literature is discussed and different problems arising are classified. At the end of section 2 a generic allocation model is described, covering the different requirements of different workcenters (side constraints and objectives) in a modular design principle. Explicit models are not discussed (the model is a superset of single solutions presented in former papers). Some application scenarios are given in section 3. In section 4 the generic allocation concept is discussed from the viewpoint of system architecture and IT requirements. In section 5 we give a conclusion and an outlook.

2 A GENERIC ALLOCATION CONCEPT

2.1 Optimization of Work Centers

A very high similarity of problems can be recognized while looking at different semiconductor workcenters. The complexity is avoiding efficient handling of the overall problem in one single mathematical model due to the number of variables. This is the motivation for decomposition of problems as presented for the lithography area in (Klemmt et al. 2010). The same criteria also apply for other workcenters. Long term capacity optimization does not include as many constraints as real-time lot scheduling on the shop floor, where scheduling does not lead to significant improvements if capacity allocation is not optimized. The way of decomposition as shown for the lithography area is also valid for other workcenters. Therefore, we established a generic multi-stage optimization approach consisting of four stages which is shown in Figure 1.

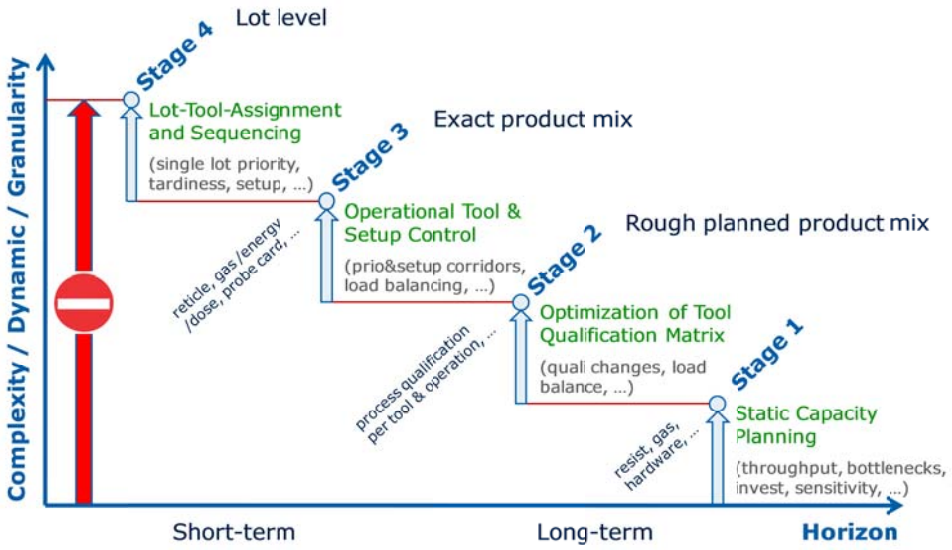

Figure 1: Workcenter problems an granularity levels – a decomposition

In every stage different objectives are optimized with regard to different kinds of constraints, data granularity and planning horizons. The optimized bounds of the previous stage are forwarded as constraints to the next stage. Each stage limits the degree of freedom and thus the complexity of the solution process. In the first two stages we find a robust tool allocation which is suitable for balanced utilization of the resources for the next weeks on a product group level (e.g. litho cluster, resists). There are two major steps to be solved: Number of resource combinations to be installed and what product to run on which

hardware. In the next two stages the detail level is increasing up to lots and assigning the resources to the specific volume slots. The calculated optima from former stages set the bounds for these allocation steps.

2.2 Modeling and Case Coverage

Typically, different classes of mathematical models are used for modeling the problems of the stages 1 to 4. While static capacity planning can be covered by linear programs (LP) and quadratic programs (cf. Harrison, and Williams 2007; Gold 2004) the problems in the stages 2 to 4 often require integer constraints, too. This is caused by the granularity level and several side constraints. For static allocation problems with integer constraints mixed integer programs (MIP) are used primarily. Static allocation means, that the model output is valid for a predefined time horizon – without explicitly modeling a time axis in the model. In stage 4, the scheduling level, it is crucial to reflect dynamic aspects by modeling the time axis. This leads to disjunctive models which can be covered – in limited problem dimensions – by constraint programs (CP) or mixed integer programs (cf. Klemmt 2012). The limitation results from the NP-hardness of most scheduling tasks (Brucker 2004; Pinedo 2008). Figure 2 drafts this classification on a very simple level.

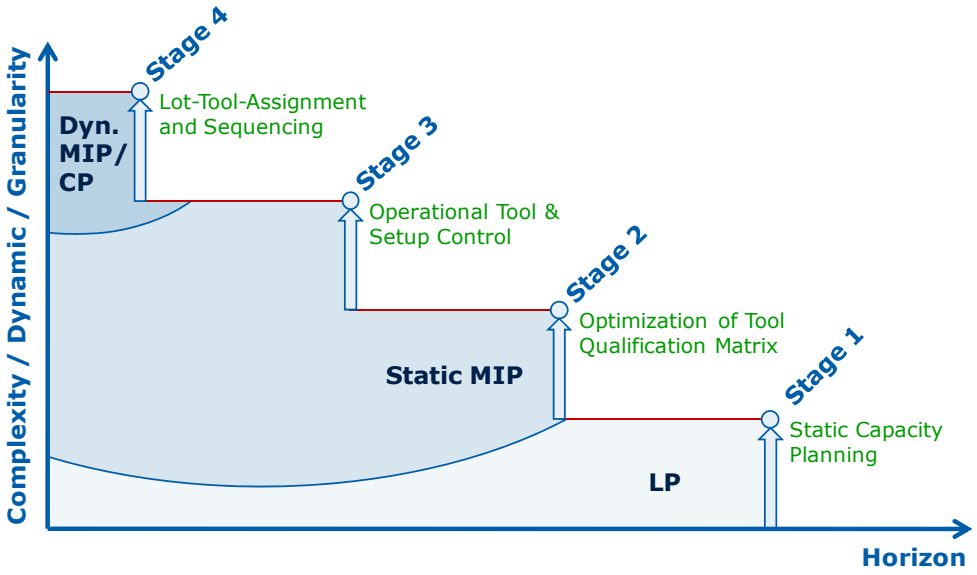

Figure 2: Classification of mathematical models for workcenter problems

In this paper a generic allocation concept for the problems of stage 1 to 3 is discussed. It is based on a static MIP model which naturally covers the LP-level. Scheduling problems are not covered. However, as mentioned in section 2.1 all decisions in stage 4 implicitly depend on the results of stage 1 to 3.

2.3 The Unified Model Concept

The basic question which has to be answered by a static allocation problem is: "How many units of a 'job class' have to be assigned to a 'resource' with regard to a given set of constraints in a predefined time horizon, fulfilling an objective in sense of optimality?"

The crucial point in this question is the interpretation of what is a 'job class' and a 'resource' and the definition of the time horizon. This question has to be answered by each application, using the generic allocation concept, itself. The mathematical model only knows the vocabulary: job class, resource, setups and secondary resource. It also provides a large set of possible constraints and objectives bringing this vocabulary in connection.

Now we want explain the definition of job class and resource on the example of static capacity planning (stage 1) in Figure 3. A product mix is given by a demand, which is assigned to a set of routes. Each route consists of a sequence of processing steps. Some of these (single process) steps are very similar because they have the same pattern of capacity consumption regarding the 'resources' (e.g. tools). For instance it is not always necessary to reflect all different products and steps, because some routes will have the same specifications (dedications) for some of the processes. So, by combining them into ' job classes' we are downsizing the capacity planning problem. The result of optimizing the problem drafted in Figure 3 is a tool (='resource') utilization profile with (145h; 138h; 145h; 160h) – if the objective is load balancing. In static capacity planning the time horizon is implicitly defined by the demand – typically wafer starts per week. For more details we refer to (Klemmt, Laure, and Romauch 2012).

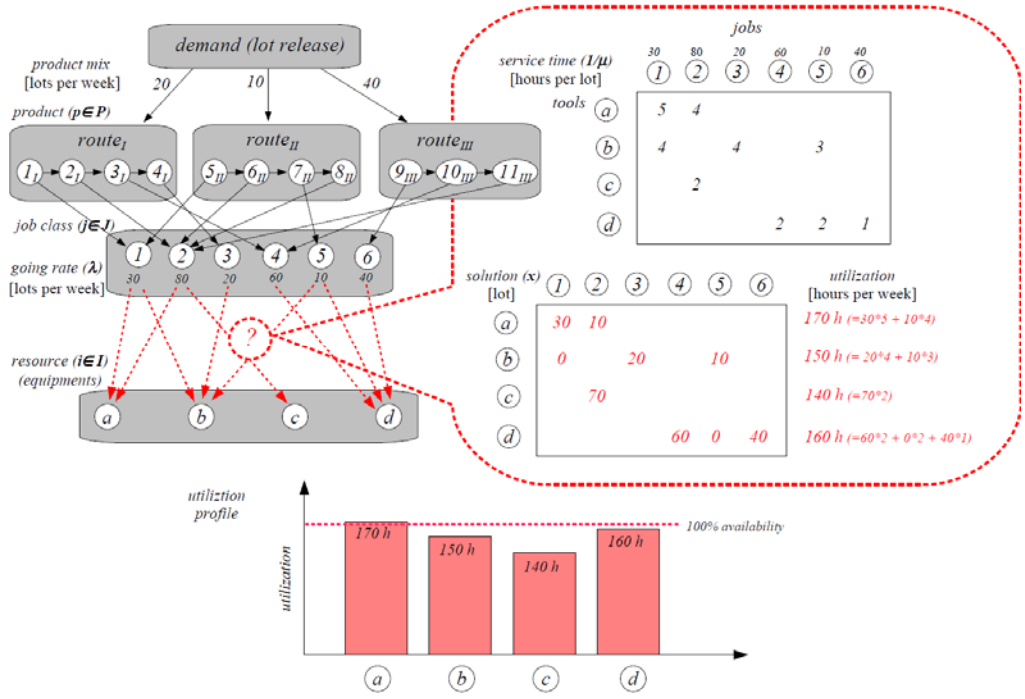

Figure 3: Example static capacity planning (cf. Klemmt, Laure, and Romauch 2012)

Now, we can use a generic allocation model also in another context: e.g. for reticle assignments (stage 3). Here, we are on the operational level and the task is to assign all lots currently waiting at the lithography operation to a set of tools (foto clusters). For processing a lot on a tool a specific reticle (mask) is needed, which is moveable between the tools. In this example the 'job class' is given by the reticle and the demand is the amount of lots standing behind the reticle. The objective – next to load balancing – is for example the reduction of reticle moves. The (valid) time horizon of this allocation is consequently significantly shorter than in static capacity planning – typically one hour. For more details we refer to (Klemmt et al. 2010).

There are several other problems too which can be modeled on this abstract level. A 'job class' can also be a probecard (cf. Klemmt, and Weigert 2011) in wafertest, an operation for the planning of tool qualification matrices (cf. Klemmt et al. 2010) or even a single process step. The demand is the volume behind this aggregation level. A 'resource' is not necessarily restricted to physical equipments. It can also model combinations of tools and secondary resources. On this level of abstraction, a large set of constraints is defined, influencing a job class-resource-allocation.

Typical constraints are:

- Dedication constraints (allocation allowed or not),
- Heterogeneous processing times (within a workcenter),
- Min./max. constraints concerning units and load (load=unit*time) per allocation,
- Counter constraints (min./max. number of allocations per resource/job class/secondary resource),
- Setup constraints (several job classes requiring a specified tool state),
- Mapper constraints (an allocation requires additional (limited) capacity; e.g. secondary resource),
- Excluding allocations (forbidden combinations/sequences of job classes/setups), …

Also a wide range of objective functions is defined, measuring the performance of an allocation. Typical objectives are:

- Minimization of the maximum load on a resource (load balancing),
- Minimization of setups (total sum or maximum number of setups on a resource),
- Costs (setup cost, allocation costs), … .

Now, all constraints and objectives are available in sense of a modular design principle. So, each specific application defines itself if a constraint is active or not. Thereby, an activation of a constraint requires some inputs (data, mappers) in a predefined common data layer structure. Furthermore, the application defines an objective. If more than one objective is selected a multi-criteria objective function can be defined (weighted sum) or an objective hierarchy is defined. In the last case, the problem is than solved iteratively in a cascading way (optimization concerning objective 1; fix objective function value 1; optimization concerning objective 2; …) – cf. (Klemmt et al. 2010).

In Figure 4 the generic allocation concept is drafted on the basis of several applications which are reducible to static allocation problems.

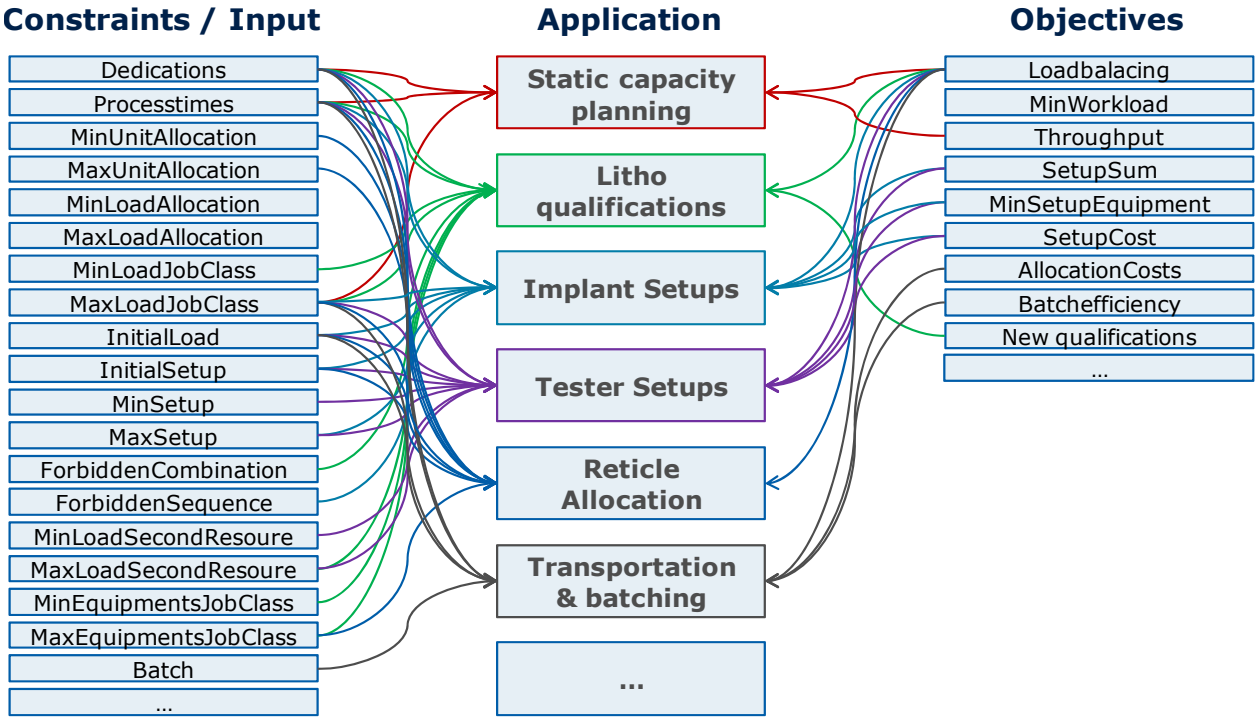

Figure 4: Generic allocation concept

3 APPLICATION EXAMPLES

3.1 Current Status

The potential of solver based applications to improve productivity was already shown at Infineon. There are several use cases at shop floor control and planning level where system complexity could be reduced in parallel:

Capacity Planning	Load balancing in complex dedication scenarios has been a weakness of planning systems formerly used. It had to be configured manually and did not provide optima. Meanwhile, optimization has become a standard feature in our system and the LP solver based functionality (cf. Gold 2004) could be covered by the generic allocation component.
Tool Qualification	Optimization and adaptation of workcenter configurations according to product mix changes often has been a trial and error approach based on static models. We implemented a solver based solution for lithography which allows to optimize resist allocation and tool qualification in one procedure based on controllable objectives (cf. Klemmt et al. 2010). It includes the usage of open source and commercial products alternatively. The generic allocation approach will enable easy transfer to comparable use cases.
Reticle Allocation	Dispatch rules for lithography became quite complex over time considering all parameters necessary for lot sequencing, reticle changes, R2R and split/merge control. Thus its load balance feature was limited and required additional monitoring at shop floor level. Assigning reticles (lot groups) to available equipments based on constraints and delivery predictions can be modeled as a static allocation problem (cf. Klemmt 2012) where we applied the generic approach described in this paper.
Setup Control IMP	Setup rules considered parallel machine status in order to reduce setup times and limit queue length while preventing inappropriate setup combinations. However, there was still a gap to a fully automated mode considering critical changes at certain equipments. The reticle allocation model mentioned above can easily be extended to setup optimization problems at the implant area. Real data models are running.
Setup Control Test	Short term setup control for functional wafer test area is challenging due to reentrant flows. Time consuming manual planning with limited accuracy needed to be replaced. Combining a MIP allocation model and a discrete event simulator within an iterative optimization procedure allowed us to benefit from short static model calculation times while, on the other hand, providing a lot based schedule (Klemmt, and Weigert 2011).

Transport Control	Besides stocker capacity and equipment availability rolling horizon parameters were calculated in order to adapt volume-to-capacity balance between alternative production areas. This lot transport destination settings showed limitations at e.g. batch operations and a more precise capacity check was required in order to reduce moves between buildings. Using the generic allocation component we generate lot assignment proposals which consider all relevant tool specific constraints. Thus, we are able to reduce long distance transports significantly in pilot areas.

3.2 Future Work

There is potential to further utilization of the flexibility of the generic allocation concept in many ways. Future use cases for static model applications may include:

- Advanced approaches to consider quality related tool preferences,
- WIP balancing across closed machine sets or automated backup tool release at bottlenecks,
- Cluster tool feeding based on availability of chamber combinations,
- Batch building at small lot sizes considering limited tool internal carrier storage capacity,
- Capacity check at downstream equipments for time constraint based lot release,
- Improved low volume and priority lot-to-tool assignment.

Most of these scenarios would improve existing solutions and/or reduce system complexity if the generic approach is used. Section 4 will show a basic strategy how to integrate the unique model into a framework.

4 FRAMEWORK APPROACH

4.1 Requirements to System Architecture

The implementation of mathematical optimization software plays a crucial role for the manufacturing organization. People on the shop floor and in the planning department rely on their daily operations or business decisions. For this reason, correct and consistent behavior of the software systems is a fundamental part of the end users expectations. On top, the management requires cost-effective development, maintenance, and operation of the software.

To implement mathematical optimization software as additional components in an existing fab MES system architecture several challenges have to be solved by IT. Especially in high automated fabs software components should follow the same requirement specifications like standard process equipment, e.g. reliability, availability and maintainability (RAM). The following detailed requirements for the new software were given by the IE and production department:

- The software has to be implemented in different wafer fabs with different MES system architectures,
- The software will be used in full automated fabs as well as in manually operated fabs,
- The software has to support on demand decisions at the shop floor and real time decisions,
- The software should be reusable for allocation problems,
- The optimization engine should be build in as a plug-in component to have the flexibility for further development,
- A fall back solution is required in case the optimization engine will not provide a solution in a timely manner,

- What-if analysis capability is required to analyze changes prior to their implementation,
- Optimization results should be validated easily and verified by the end users,
- The execution of the results should be monitored with actual data from the shop floor.

To handle the complexity of this application and their maintainability a framework approach was developed. Maintainability is defined as how easy changes can be made to the software components. This includes changes for adaptation of the system to meet new requirements, changes for additional new functionalities and changes for corrections when deficiencies occur.

4.2 Steps to a Fab Solver Framework

The basic Fab Solver Framework approach is shown in Figure 5. To integrate the solver in different computer integrated manufacturing (CIM) environments as a first step a Common Input Data Layer was defined. As described previously most of the solver applications require the same input data structure or a subset of them. For each application of the scheduler an input data adapter has to be built to collect the data from the site specific data sources. The adapter has to verify and validate the data collection, has to control the frequency of the data update and has to identify the data for the different applications.

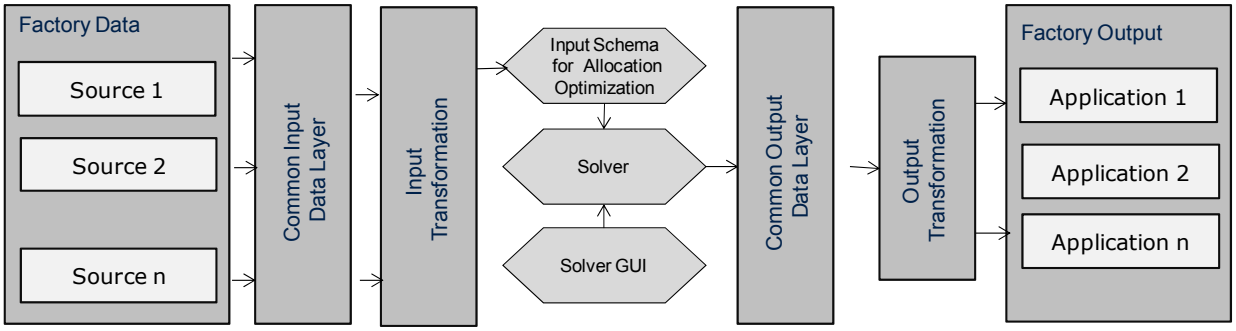

Figure 5: Fab Solver Framework

In a second step a Solver Input Transformation will be performed which translates the data from the Common Input Data Layer to the solver specific input matrixes and vectors. This Solver Input Transformation Component is specific to a commercial or an open source solver software and to the mathematical problem description. As pointed out, if most of the solver application can be represented by only one problem description, the complexity of this component can be reduced greatly. Only this component needs to be extended if more than one solver software is used or a second mathematical problem description is developed.

In a third step the actual optimization has to be executed. Therefore a User Interface has to be developed to parameterize the constraints, to define the object function and to define runtime parameters for the optimization. Part of the User Interface is a role concept to allow a differentiation between administration and user rights. Because the execution time is crucial for many applications a client/server architecture was developed. With the client/server architecture the optimization runs can be performed on high performance computers. It provides a cost efficient implementation and is scalable for future performance requirements.

In a last step the results of the solver are stored in a Common Output Data Layer. From there, an output transformation can be performed which formats and provides the results according to the target application. For example the results can be visualized in some information systems to support manual decisions or they can be used by a dispatching systems or can be integrated in automation workflows.

The framework provides a flexible solution to support the multi-stage optimization approach. Based on the applications the framework can be integrated in the planning environment, in different MES environments or can be integrated with discrete event simulation software to perform what-if analysis.

5 CONCLUSION

Mathematical optimization has become more and more important in several areas of semiconductor manufacturing. In the past, we have designed many single applications to specific problems with very good results. Now, we faced the variety of the applications to be maintained, where the core of each application is similar to each other. Therefore, we forced the successful development of a generic model approach that is able to solve different problems. The basic mathematical model uses a mixed integer program with dynamically linked constraints embedded in a framework that clearly defines the task of each functional layer. Single applications were quickly transferred to the new generic framework approach. The numerous optimization demands in all areas of the fab will lead to a fast extension of applications covered by the framework. We are very encouraged to pull in more tasks to the solver approach to reduce complexity of customized state-of-the-art solutions on the one hand and to improve target achievement close to optimum on the other hand. The future work will focus on the extension of the generic mathematical model as well as on the improvement of the system interfaces to achieve a plug-and-play solution for many use cases.

ACKNOWLEDGMENTS

This work was supported by the Federal Ministry of Education and Research of Germany (promotion number 13N11588).

REFERENCES

Akcali, E., A. Üngör, and R. Uzsoy. 2005. *Short-Term Capacity Allocation Problem with Tool and Setup Constraints*. In *Naval Research Logistics* 52:754-764.

Bixby, R., R. Burda, and D. Miller. 2006. "Short-interval detailed production scheduling in 300-mm semiconductor manufacturing using mixed integer and constraint programming." In *Advanced semiconductor manufacturing conference*. 148–154.

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.

Chung, S.H., C.Y. Huang, and A.H.I. Lee. 2006. *Using Constraint Satisfaction Approach to Solve the Capacity Allocation Problem for Photolithography Area*. In *Computational Science and Its Applications* 3982:610-620.

Doleschal, D., J.Lange, and G. Weigert. 2012. "Mixed-integer-based capacity planning improves the cycle time in a multistage scheduling system." In *Proceedings of the 22th International Conference on Flexible Automation and Intelligent Manufacturing*.

Gold, H. 2004. "Dynamic Optimization of routing in a Semiconductor Manufacturing Plant". In *Operations research proceedings 2004: selected papers of the annual international conference of the German Operations Research Society (GOR);* jointly organized with the Netherlands Society for Operations Research (NGB), edited by P. K. Hein Fleuren, Dick den Hertog, 76–83. Tilburg, The Netherlands: German Operations Research Society (GOR) and the Netherlands Society for Operations Research (NGB).

Harrison, J. M., and R. J. Williams. 2007. "Workload Interpretation for Brownian Models of Stochastic Processing Networks". In *Mathematics of Operations Research*, 32 (4):808–820.

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

Klemmt, A., and G. Weigert. 2011. "An optimization approach for parallel machine problems with dedication constraints: Combining simulation and capacity planning." In *Proceedings of the 2011 Winter Simulation Conference*, 1986-1998.

Klemmt, A. 2012. *Ablaufplanung in der Halbleiter- und Elektronikproduktion: Hybride Optimierungsverfahren und Dekompositionstechniken*. Springer Vieweg.

Klemmt, A., W. Laure and M. Romauch. to appear 2012. "Product Mix Optimization for a Semiconductor Fab: Modeling Approaches and Decomposition Techniques." In *Proceedings of the 2012 Winter Simulation Conference*.

Mönch, L., J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, and O. Rose. 2011. "A Survey of Problems, Solution Techniques, and Future Challenges in Scheduling Semiconductor Manufacturing Operations." *Journal of Scheduling,* 14(6):583-599.

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

Toktay, L.B. and R. Uzsoy. 1998. *A capacity allocation problem with integer side constraints*. In *European Journal of Operational Research* 109:170-182.

AUTHOR BIOGRAPHIES

SEBASTIAN WERNER Sebastian Werner obtained his masters degree in Electrical Engineering in 1997. He was a Research Assistant at the Centre of Microtechnical Manufacturing of the Dresden University of Technology until 2005. He works as Senior Manager in the Industrial Engineering department of Infineon Dresden. His email address is Sebastian.Werner@infineon.com.

FRANK LEHMANN received his masters 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.

ANDREAS KLEMMT received his master's degree in Mathematics in 2005 and Ph.D. in Electrical Engineering in 2011 from the Dresden University of Technology. He works as Operations Research and Engineering Expert at Infineon Dresden. His current research interests are mathematical programming, capacity planning, production control and simulation. His email is Andreas.Klemmt@infineon.com.

JOERG DOMASCHKE received his master degree in Computer Science at the University of Wuerzburg. He works as a Manager in the Manufacturing IT department responsible for WIP Flow Management. In the past he worked in different positions in the Area of Operations Research and Lean Manufacturing. His email address is Joerg.Domaschke@infineon.com.