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

FROM DISPATCHING TO SCHEDULING: CHALLENGES IN INTEGRATING A GENERIC OPTIMIZATION PLATFORM INTO SEMICONDUCTOR SHOP FLOOR EXECUTION

Andreas Klemmt
Jens Kutschke
Christian Schubert

Infineon Technologies Dresden GmbH
Königsbrücker Str. 180
01099 Dresden, GERMANY

ABSTRACT

This paper shows how automatic optimization of scheduling problems has been integrated within the automation framework of a semiconductor factory. Special attention is paid to the requirements arising from such an application in real world production in terms of constraints and objectives as well as from a factory integration perspective such as autonomous operation, high availability and efficient maintenance. Subsequently, possible solutions on how such requirements can be addressed will be discussed. Thereby, the advantage of using Constraint Programming solvers is highlighted. Knowledge that was gained during the implementation is presented for selected cases followed by the benefits that were achieved.

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 who do 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 solutions. Although standard dispatching systems have been pushed to their limits in terms of modeling capability and response times, there is still a gap of productivity to utilize which may become remarkable.

Meanwhile, mathematical programming as an alternative optimization technique plays an important role in the semiconductor industry (cf. Bixby, Burda, and Miller 2006; Mönch et al. 2011; Klemmt 2012). It provides objective functions and constraint based self-adapting systems. Manual efforts in rule adaptation are significantly reduced because no “strategy” has to be modeled explicitly. Since powerful commercial scheduling solutions are available, Infineon decided in 2012 to begin a standardization process for these kinds of applications (Werner et al. 2012). Five years later, a framework has been developed which is able to deal with different work centers’ requirements (operations, equipment, constraints & objectives) and is used over all frontend and selected backend sites.

This paper is a follow up to (Werner et al. 2012) and is organized as follows: In section 2 we highlight situations which are difficult to address using classical dispatching. Section 3 discusses important features of the employed model, in particular time models as well as constraints and objective functions. The model has become part of a framework which is describes in section 4. It is followed by a discussion of the knowledge that was gained during the implementation of scheduling on the shop floor. The benefits that could be achieved are drafted in section 6 leading to the conclusion in section 7.

2 DISPATCHING VS. SCHEDULING

The most common solution to assigning jobs to specific machines in industry is the use of dispatching algorithms. Dedicated software packages, like RTD (RealTime Dispatcher) are readily available for this purpose. Dispatching rules, in general, decide what to do next, only based on current information on jobs and tools without considering every possible future combination. Thus, they essentially provide heuristics to the underlying highly complex optimization problem. In contrast to scheduling, dispatching is a well-adapted technique in industry, since:

- it is a very intuitive approach to the problem,
- complex requirements can be implemented,
- the computational effort is small and
- acceptable results can be achieved in a short amount of time.

As already stated by (Fordyce, Bixby, and Burda 2008) the usage of scheduling enables us to

- look across time,
- look across tools at a tool set,
- create an anticipated sequence of events at a tool set over some time horizon,
- establish a formal metric and
- search alternatives

For more fundamentals in dispatching and scheduling we refer to (Pfund et. al 2006). In the remainder of this section typical situations, that are often observed in practice, will be shown which shall further motivate the usage of scheduling.

2.1 Balancing

In semiconductor manufacturing a pool of machines which are able to achieve the same process results is called a work center or closed machine set. However, not all machines within such a work center are qualified to perform every process due to different capabilities of machines, the high number of different processes and the corresponding effort to ensure that their process results are within tight specifications. This leads to a problem of unrelated parallel machines with machine eligibility restrictions (Pinedo 2016). Especially in high mix wafer fabs, the dedication matrix M, whose elements m_{jk} are one if a job j may be run on a machine k and zero otherwise, takes on a sparse structure. A simple, well-known example where dispatching may lead to unbalanced machine utilization is depicted in Figure 1.

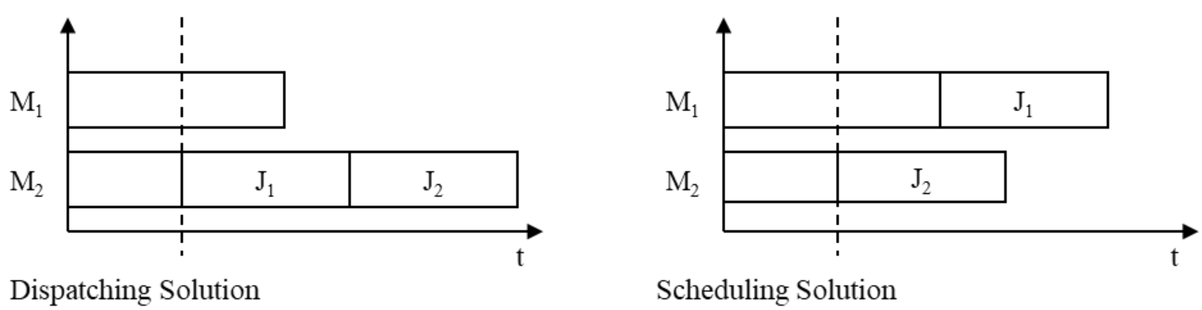

Figure 1: Dispatching vs. Scheduling Solution, J_2 dedicated to M_2, $Rank(J_1) > Rank(J_2)$.

Here, the dispatcher decided to assign the higher ranked J_1 to M_2. Since J_2 dedicated to M_2, it must wait for completion of J_1 before it can be processed resulting in a sub optimal makespan. Given equal weights for the cycle times of both jobs, a scheduler may choose to delay J_1 in favor of the lower ranked J_2 leading to a more even machine utilization.

2.2 Unrelated Machines

Often a work center comprises unrelated machines leading to different process times and different minimum or maximum batch sizes. This also leads to situations where dispatching yields non-optimal solutions, since it may be beneficial to deliberately delay a single higher ranked job in order to finish other jobs earlier consequently reducing the makespan while increasing batch efficiency, cf. Figure 2.

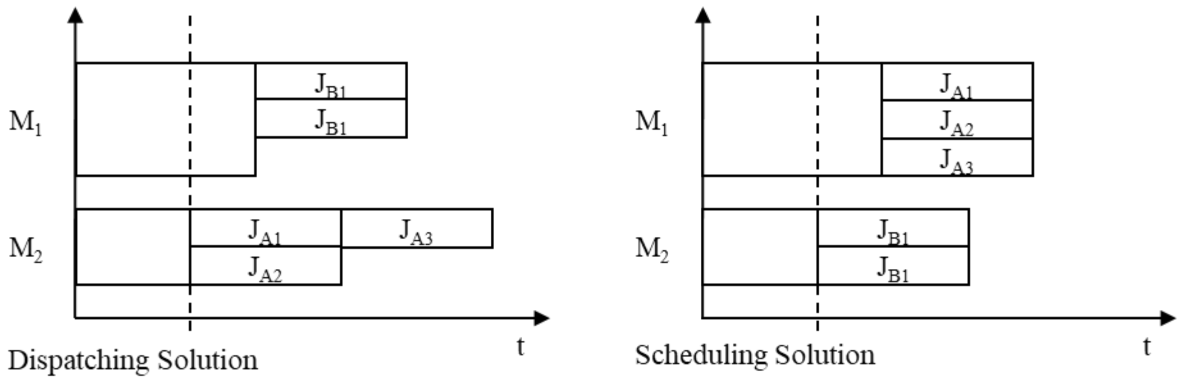

Figure 2: Possible results for different batch sizes, job families A & B, Rank(JA_k) > Rank(JB_r).

2.3 Time Related Constraints

Time related constraints are difficult to consider in a dispatching environment. A common approach is to consider jobs with time constraints to be of a high priority class. However, this may negatively impact throughput as well as delay actual high priority jobs.

A scheduling approach is able to outperform a dispatcher with respect to time related constraints as it has access to the time axis and can thus evaluate them when finding an optimal solution.

2.3.1 Hard Time Related Constraints

A characteristic feature of semiconductor manufacturing is the presence of time bound constraints at certain steps. A common source of such time bound constraints are cleaning/etching steps leaving the surface area susceptible to oxidation. To minimize deterioration a time bound constraint to the next oxidation/deposition/diffusion step is imposed by quality engineers. Violation of such a time bound, usually leads to the wafers being scrapped and consequently to a considerable financial loss. Thus, time bounds are considered to be hard time constraints (cf. Klemmt and Mönch 2012).

Another time related constraint might arise from adhering to a given due date. This is of special interest when dealing with batch processing machines where large batch sizes shall be achieved.

2.3.2 Soft Time Related Constraints

The class of soft time related constraints comprises constraints whose violation does not cause scrapping of wafers, but may have a significant negative impact on performance. Thus, machines are monitored on a regular basis to ensure high quality. Not every quality indicator can be measured in situ, i.e. directly on productive wafers. Therefore special test runs are performed within regular intervals yielding due dates for test wafers. Missing such a due date incurs a high quality risk und ultimately leads to blocking the process on that specific machine and consequently worsens machine utilization.

In addition machines can be used to produce test wafers for other machines. If such a job is not finished on time, it may cause the problems described above for the receiving machines. However, from the company's point of view, test wafers do not add value as they cannot be sold. Hence, a good tradeoff has to be found between meeting due dates of test wafers and not increasing cycle times of development and production wafers.

Another source of soft time related constraints are machines that require conditioning, i.e. after running a conditioning job, a fixed time window starts in which productive lots may be assigned to this process. For some machines, the validity of a conditioning is not related to a time window but to a counter which is reset after conditioning and decreased by every wafer that is being produced. A new conditioning is required as soon as the counter reaches zero.

Some processes, especially in implant work centers, may exhibit the opposite behavior. There a process may only be run for a given amount of time, due to technological reasons. However, changing the process may incur additional setup times.

Within a dispatching environment, it is feasible to meet all these requirements. However, we observed that there was the potential for increased efficiency. Without the ability to look across time and plan in the future, it is nearly impossible to leverage this potential.

2.4 Auxiliary Resources (Durables)

In many work centers, especially Lithography and Testing, processing a job does not only require a machine to be available but also auxiliary resources, often referred to as durables. In semiconductor industry auxiliary resources comprise lithography masks/reticles, probe cards, handlers, load boards, etc.

Auxiliary resources increase complexity considerably due to their combinatory nature. The problem becomes even more involved if there are dedications and limitations between the different resources existing. Figure 3 drafts an example of six different possible ways to combine durables for processing/testing a lot in a step at an equipment. Thereby, the test itself is dedicated to equipment T1 or T3. Furthermore, a synchronization between auxiliary resources is required, as the processing can only begin as soon as all resources are available. Often additional setup times before and after processing may have to be considered, e.g. changing masks in lithography tools. Assigning a resource too early will consequently have a negative impact on performance. Many auxiliary resources are also non-stationary which requires them to be transported to the machine. Hence, transportation times have to be considered as well in this case to achieve a good synchronization.

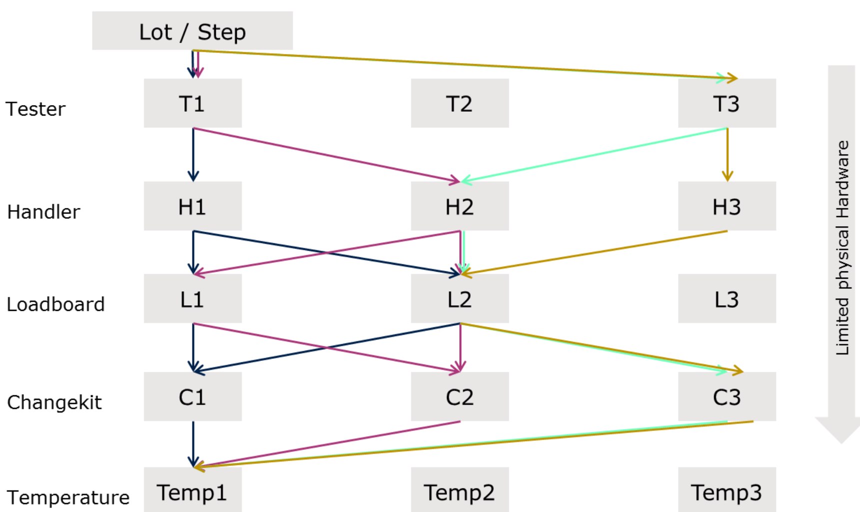

Figure 3: Interdependency between auxiliary resources in the testing area.

From past experiences we found, that dispatching was able to deal with one additional auxiliary resource but performance suffered in more complex situations (cf. Figure 3).

2.5 Advanced WIP Flow Constraints

Up to now, only requirements regarding individual work centers have been considered. Achieving good solutions on factory level, however, requires communication between work center models.

A good example are time bounds, see section 2.3.1. The work center where time bounds originate needs to take remaining capacity of the succeeding work centers into account. For example in case of a temporary tool down, the supply of jobs, equipped with time bounds, has to be reduced accordingly - otherwise, time bounds could be violated. The problem becomes even more complex in case of time bounds that span more than one work center or nested/overlapping time bounds.

3 MODELING

As pointed out in the previous section, there are numerous situations where a scheduling approach could help to improve work center performance significantly. This has led to an successive implementation of scheduling to work centers which provided the highest potential for improvements. In the following we will point out the challenges that had to be overcome.

3.1 Approaches

As shown in (Klemmt 2012; Werner et al. 2012) scheduling can be employed at different stages, cf. Figure 4. The scope of this paper is restricted to *Stage 4: Lot-Tool-Assignment and Sequencing*. Back then it was noted that there are two prevailing approaches: Mixed Integer Programming (MIP) and Constraint Programming (CP). While research mainly focusses on MIP formulations a few authors also considered the usage of CP. With the latest advances in CP solution algorithms and increasing availability of computational power, CP is used at Infineon predominantly during the last years. This approach has recently also been favored by other authors (Ham and Cakici 2016, Ham and Fowler 2017). For scheduling problems CP offers several advantages over MIP formulations:

- CP is more natural due to readily available constraints on interval decision variables
- CP allows modeling of more involved objectives (e.g. tardiness spread) and constraints (e.g. cumulative bounds, state functions, alternatives and precedence relations)
- CP has an increased readability (less need of boolean indicator variables and BigM formulations)
- For larger instances and a given a fixed amount of time: CP solvers generate better solutions

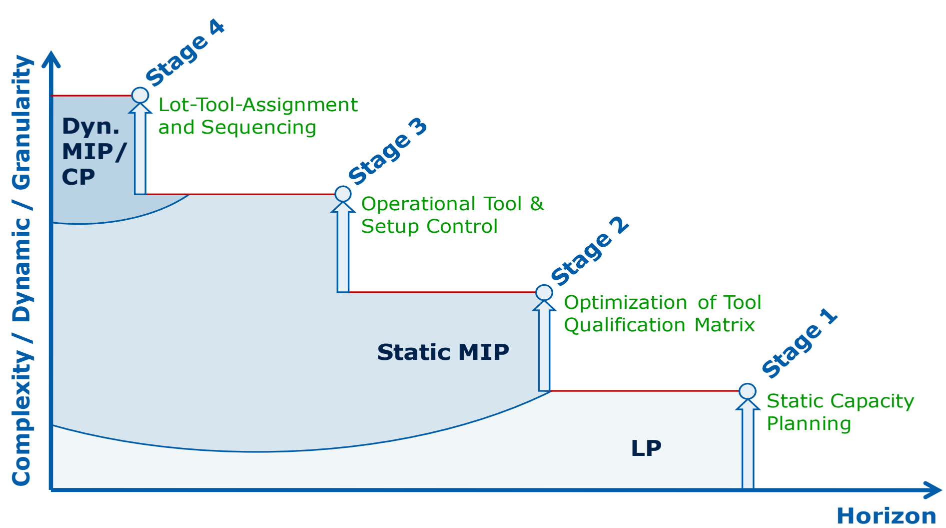

Figure 4: Classification of mathematical models for work center problems (Werner et al. 2012).

3.1.1 Remaining Runtime

The remaining runtime of a machine, on the other hand, should be as exact as possible since it directly relates to the release date of each job. Many machines also implement a queueing mechanism. This means they are able to preload a job ahead of its actual start inside the process module. This allows for a very efficient job swap minimizing idle time of the process module. Therefore a precise and reliable prediction of the remaining runtime is required.

In order to obtain exact runtime information, depending on the machine type, a discrete event simulation has been set up which mimics the internal movements of wafers and handling systems.

3.1.2 Look Ahead

Many work centers, e.g. furnaces, consist primarily of batch machines. In this environment it is highly advantageous not only to consider jobs that are already waiting inside the queue, but also jobs that are about to arrive inside the queue. These jobs can then be given corresponding release dates r_j. Knowledge about release dates enables us to actively delay jobs in order to increase batch sizes (Fowler 1992).

In this setting, release dates can only be estimated by a look ahead, since they are still in process or still in queue at previous, and not necessarily adjacent, steps. In order to arrive at a reasonable time estimate the remaining runtime as well as possible queueing and transportation times have to be estimated. Clearly, the larger the look ahead in units of time, the larger the uncertainty. Moreover, the total number of jobs increases as well, putting a heavy burden on computational effort. Thus a tradeoff has to be found.

In (Klemmt 2012), it has been shown that the best tradeoff is to predict the arrival of jobs within the time window of 50% of the average process time. Using statistical data it is possible to implement a look ahead of a given number of process steps and to predict their arrival time.

3.2 Primary Objectives

In literature one often finds that minimizing the Total Weighted Tardiness (TWT) $\sum w_j T_j$ is used as a single objective. In practical situations this often does not suffice to arrive at generally accepted schedules. While Total Weighted Tardiness is a well suited measure to achieve on-time delivery, it does not take additional goals like the reduction of cycle times as well as machine utilization into account. Figure 5 illustrates a simple example, where both schedules yield the same value of the objective function. The right schedule, however, is clearly to be preferred as it gives a smaller cycle time for J_2.

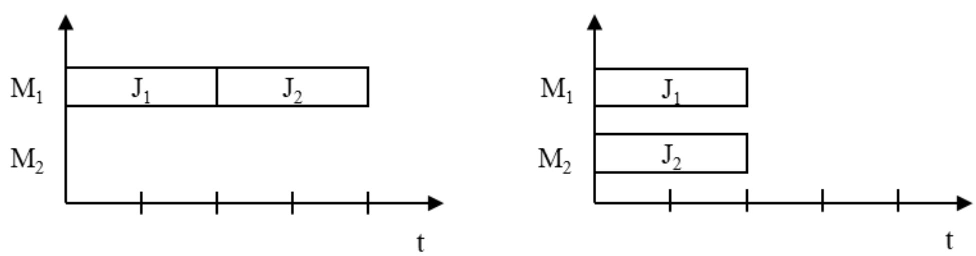

Figure 5: Schedules with same TWT (d_1=2, d_2=4) where the right schedule is preferable.

Depending on the application, it is beneficial to extend the objective function by Total Weighted Completion Time (TWC) $\sum w_j C_j$, (hard/soft) time bound violation (cf. section 3.4) and tardiness spread goals. In case of TWC, jobs with high priorities can be given a high weight, which will lead to a privileged treatment without the need of being tardy themselves.

3.3 Secondary Objectives

3.3.1 Time Bounds

As described in section 2.3, jobs are often connected with time bounds. Integrating time bounds as a constraint to the model has proven difficult since there might arise undesirable situations where it is impossible to satisfy all time bounds. In order to arrive at a solution where as many time bounds as possible are met, the violations of time bounds are added to the objective function with a high weight.

3.3.2 Virtual Time Bounds

Additionally, the concept of virtual time bounds has been introduced into the model. Virtual time bound violation is added to the objective function with high but smaller weight than real time bounds.

Every priority lot is given a virtual time bound to ensure processing within a narrow time frame. Moreover, virtual time bounds are imposed on test wafers, as described in section 2.3.2, to guarantee periodic monitoring of the machines. Also jobs whose lead time surpasses a given threshold, depending average process times and loading of the work center, are equipped with a virtual time bound to not sum up too much cycle time for individual jobs.

4 OPTIMIZATION FRAMEWORK

While constraints and objects may vary for different work centers, the process of generating and executing a schedule follows a common pattern as outlined in Figure 6.

Data required for decision making is pulled from a variety of sources (Data Acquisition) and then processed further to formulate the problem to be solved (Preprocessing). Data availability, quality and validity is key – the preprocessing therefore realizes not just a data transformation, but also a comprehensive validation as well as automatic completion and correction where possible. Transformed data is then provided to the mathematical model for execution by commercial CP solver implementations (Solution). Validation is applied on the solver results, making sure they are executable on the shop floor. In a Postprocessing phase, those results are transformed back into real world entities for later execution, which is a responsibility of existing execution (MES) or dispatching systems.

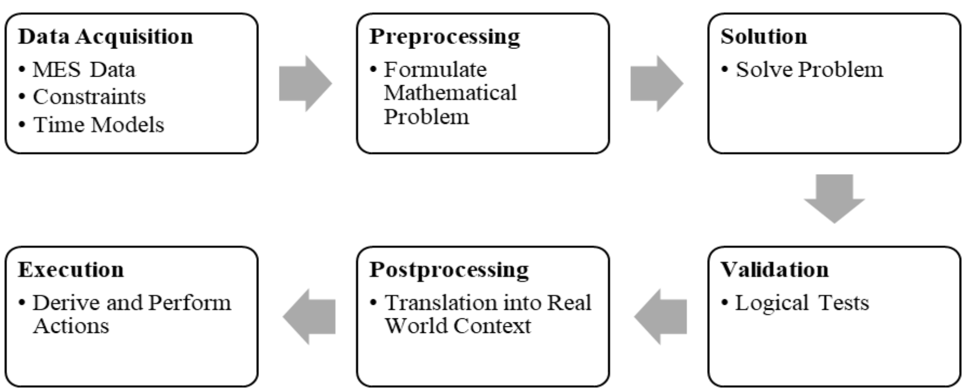

Figure 6: Steps in Solution Procedure.

This technical workflow is typically executed at fixed intervals. The length of these intervals depends on individual work center characteristics and might range from two up to 15 minutes. Therefore, no specific handling of exceptional events (e.g. unplanned equipment downs) is required – subsequent runs will automatically factor in such changes in boundary conditions.

In order to reduce efforts for developing new scheduling solutions as well as ensure an efficient long term enhancement and maintenance process, common functionality has been encapsulated as a generic framework. Some key aspects will be outlined in the following subsection.

4.1 Development and Operations

Scheduling applied to shop floor control poses some specific challenges with regards to

- availability
- quality
- agility

of the solution. Schedules need to be provided consistently (availability) and must contain decisions that can be executed directly by operators or fab automation (quality). Changing conditions in the real world need to be quickly factored into the mathematical model as they arise (agility).

Many of these aspects can be considered solved – redundant IT infrastructure that provides failover and load balancing capabilities as well as monitoring up to the application level are standard solutions and support an early detection of potential or actual failures. In contrast, the complexity of mathematical models and the huge variability and number of scenarios represented by the input data are risks to operations. Therefore the authors decided to apply DevOps principles to the mathematical models, targeting an automated process of validating changes to the model and delivering to a number of test, staging and production environments, and to implement an additional fallback layer for uncaught exceptions (cf. section 5.2).

Using Continuous Integration, changes to a model under development are automatically detected and validated using automated tests. These tests address a broad range from simple technical validations on the artefacts (e.g. syntax) to sophisticated automatic assessment of the models decisions (e.g. given a specific input, does the output match expectations) and are enhanced as the model evolves. A model and its related artefacts can then be pushed to a target system in a fully automated fashion. Instant quality feedback and deployment automation enable quick turnaround times for new requirements.

Even with thorough validation, there might be unexpected situations in 24x7 operations that cause model failures. In such cases, an additional layer of reliability has been enabled (cf. Section 5.2).

With all these measures combined, scheduling solutions can serve high volume fully automated fabs while still supporting frequent iterative changes.

5 CHALLENGES IN EXECUTION

The following subsection shares some experiences we have made while changing the shop floor control from dispatching to scheduling:

5.1 Decisions triggered from schedules

Given an optimized schedule, actions have to be derived for shop floor execution in order to follow the plan. Depending on degree of automation in the fab this can be challenging because system boundaries are changing from one schedule to the next. Typical actions are:

- When is the next job needed at an equipment
- Which lots does this job consist of (esp. in terms of batching)
- Are there additional durables to order (reticles, loadboards,…)
- Are there additional non-productive lots/tasks to order to run the job (conditioning, test wafer)
- When potential transportation task have to be created

In fully automated fabs with negligible transportation times the execution of this a actions is observed as robust. Within the initialization of the next schedule all (wanted) decisions from the last run are already (automated) executed - the lots are in transport to or already at the equipment. Less automated fabs especially if shop floor transport and/or tool load is done manually may require more effort for a stable execution. In particular they may require freeze fence / feedback mechanisms (operator has picked the lot already? or is is still free for rescheduling?). For fabs with "longer" transportation times it has also been pointed out that it is beneficial to (re)route the "near future" planned lots to the target bay recommended by the schedule upfront and to reduce the degree of freedom (dedication - available equipment for the scheduler) to this target bay equipment in the subsequent schedules.

5.2 Fallback

There are several possible reasons that a schedule generation is failing sometimes – even within an high available IT environment (see section 4). In contrast to dispatching the potential effect on production can be higher because the scheduling model typically works on closed machine set level – not on equipment level (cf. section 2). For example a wrong designed experiment with more wafers inside the lot than the recipe allows as maximum batch size of a furnace tool will lead to a constraint violation and model infeasibility (= schedule crash) for the whole work center.

To overcome this, we have implemented a fallback mechanism. Thereby the execution system checks that the schedule is up to date and (still) valid, since machines may have changed their status since the calculation has begun. If validity is confirmed and a job is found inside the schedule that is within a configurable time range, that job is allocated to that machine within the MES system. In case no schedule has been provided, due to inconsistencies in the input data, problems IT component or infeasibilities inside the model, the execution system retreats to using dispatching rules until problems have been fixed.

5.3 Data readiness

Before introducing scheduling to a fab – especially to fabs with less degree of automation – a data readiness project has to be raised. Work methods and shop floor execution strategies have to be reviewed detailed. If there is any decision made or any constraint existing based on data which is not existing in a suitable IT system the fab is not ready for scheduling – data integration projects have to be set up and finished first. Another important point is the more sophisticated models, especially time models (cf. section 3.2) have to be defined/derived from of this data to make them usable for scheduling. Here typically the Industrial Engineering department plays an important role.

5.4 Planned and unplanned model extensions

There are several commercial products existing offering scheduling solutions. But before buying a "Black Box" there has to be check of the extendibility of the solution. We have learned from experience, that not every (especially logistical) constraint or objective is conclusively specified by the process engineer at the beginning of the scheduling project. Much can be learned from a "first" scheduling pilot leading to further input that needs to be modeled. On the other hand it has pointed out that it is beneficial to develop (enlarge) a complex model step by step. For example for a furnace we have started first with a batch work center model followed by look ahead extensions to the wet benches and actions to them, followed by some advanced modelling concerning queueing or nonproductive tasks (test lots, dummies,…).

Summarizing, we strongly recommend that the scheduling model has the capability to be open for modeling new constraints and objectives or modifying them easily.

5.5 Complexity and decomposition

Practical scheduling problems are typically NP-hard. Next to the significant performance improvements, especially for CP-Solvers, over the last years and its application to larger / practical relevant problem instances (Ham and Cakici 2016, Ham and Fowler 2017), there is still the possibility (WIP bubbles, machine breakdowns,…) that a model is leaving the typical problem dimension it was designed for. The number lots and (released) equipment is a good indicator for model complexity because they typically represent the key decision variables in the mathematical model. In practice we reduce problem complexity upfront if an exceedance of problem dimension based on that key is detected. This can be done by simply skipping some non-important lots out of the schedule, grouping "similar" lots or by more sophisticated decomposition techniques (cf. Klemmt 2012) working with time windows or dissecting the closed machine set in smaller subsets.

5.6 Training efforts

Introducing scheduling to the shop floor also has a significant impact on the line staff. The representation of classical dispatch lists is enlarged or even replaced by Gantt charts offering much more detailed information. In the past the line staff has some clear rule in mind, leading to specific decision (by ranking functions). Now the decision is derived from a more complex – for them "Black Box" – mathematical model. For example there are now situations where the scheduler will actively wait for a lot to form a bigger batch (other examples cf. section 2) but the line staff is simply seeing an idling equipment, this may lead to confusion at the beginning (is this planned or a bug?). So, the effort to train line experts in reading and interpreting Gantt charts is also a point that should not be underestimated.

6 BENEFIT

The benefits realized after introducing work center scheduling exceeded our expectations. We found that the effects vary by process as shown in Figure **7**. However, one can see that capacity has always been gained while maintaining or even reducing cycle time. This confirms the hypothesis about the benefits from (Bixby, Burda, and Miller 2006).

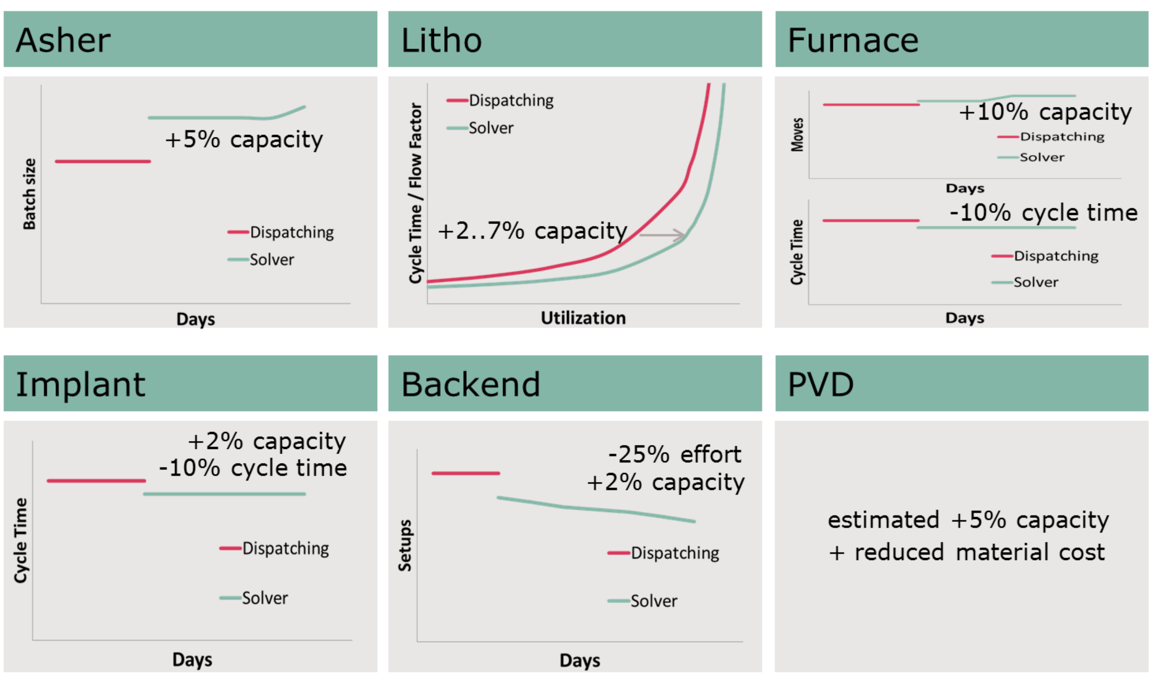

Figure 7: Benefits from Introducing Scheduling (Werner 2015).

In addition to these main KPIs, we received a lot of positive feedback from production regarding a better transparency for "the near future" which is provided by the scheduler. The Gantt charts bring additional information to the line and help to identify gaps that were not visible before. Also for cases of on call support, Gantt chart visualizations help to identify potential problems without having to debug rules or source code.

From an IT perspective the framework concept allows for a fast transferability of different applications to other sites because only parts of the workflows needed to be adapted to site specific environments, such as different MES systems. This reduces the overall maintenance efforts.

7 CONCLUSION

This paper discussed a real world implementation of a scheduling framework to solve complex flexible job shop problems from semiconductor industry. Practical situations which cannot be addressed using dispatching were highlighted, followed by a review of additional information that is required when switching to scheduling. Furthermore, special aspects of the underlying models and modeling techniques are discussed. It has been highlighted that especially CP solvers will play an important role in the future. Knowledge that was gained during the implementation was presented followed by the benefits that were achieved. In the future, we plan rollouts of the framework to further work centers and to other facilities (esp. backends and wafer tests). In addition a stronger collaboration with supply chain is envisaged.

ACKNOWLEDGMENTS

The work has been performed in the project eRamp (Grant Agreement N°621270), co-funded by grants from Austria, Germany, Slovakia and the ENIAC Joint Undertaking.

REFERENCES

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.

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*.

Fordyce, K., R. Bixby, and R. Burda. 2008. “Technology That Upsets the Social Order: a Paradigm Shift in Assigning Lots to Tools in a Wafer Fabricator - the Transition from Rules to Optimization”. In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. Mason, R. Hill, L. Moench, and O. Rose, 2277–2285. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fowler, J. W., D. T. Phillips, and G. L. Hogg. 1992. “Control of Multiproduct Bulk Server Diffusion/Oxidation Processes”, *IIE Transactions*, 24(4), 84-96.

Ham, A., and E. Cakici. 2016. “Flexible Job Shop Scheduling Problem with Parallel Batch Processing Machines: MIP and CP approaches”. In *Computers & Industrial Engineering*, Vol. 102, 2016, 160-165.

Ham, A., and J. Fowler. 2017. “Constraint Programming Approach for Scheduling Batch Operations with Incompatible Job Families in Fab and Compatible Job Families in Backend”. In *ASMC* 2017, May 15-18 2017.

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

Klemmt, A., and L. Mönch. 2012. “Scheduling Jobs with Time Constraints Between Consecutive Process Steps in Semiconductor Manufacturing”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, R. Pasupathy, and J. Himmelspach, 2173-2182. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

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.

Pfund, M. E., S. J. Mason, and J. W. Fowler. 2006. "Semiconductor Manufacturing Scheduling and Dispatching", Handbook of Scheduling,Springer International Series, 213-242.

Pinedo, M. 2016. *Scheduling: Theory, Algorithms and Systems*. Springer.

Werner, S., F. Lehmann, A. Klemmt, and J. Domaschke. 2012. "One Solver for All - a Generic Allocation Concept for Planning and Shop Floor Control". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, R. Pasupathy, and J. Himmelspach, 2124-2133. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Werner, S. 2015. "Fab Performance Increase by Using Solver Solutions". In *19th Fab Management Forum*, 05–06 October 2015, Dresden.

AUTHOR BIOGRAPHIES

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 Staff Engineer in the global Operations Research Group of Infineon.. His current research interests are scheduling, mathematical programming, capacity planning, production control and simulation. His email is Andreas.Klemmt@infineon.com.

JENS KUTSCHKE received his degree in business informatics from the Staatliche Studienakademie Dresden in 2001. He works as a Staff Engineer for Factory Integration at Infineon Dresden. His work focuses on software architecture and framework development in the context of scheduling. His email is Jens.Kutschke@infineon.com.

CHRISTIAN SCHUBERT graduated with a master's degree in Control Engineering from Manchester University in 2008 and received a German Diploma in 2009 from Dresden University of Technology in Mechatronics. He continued to work as a research assistant and got his Ph.D. in Multibody Simulations in 2014. He works in Automation at Infineon Dresden. His email is Chr.Schubert@infineon.com.