DOI: 10.22032/dbt.57896

Simulation in Produktion und Logistik 2023
Bergmann, Feldkamp, Souren und Straßburger (Hrsg.)
Universitätsverlag Ilmenau, Ilmenau 2023
DOI (Tagungsband): 10.22032/dbt.57476

To Batch or Not to Batch? Real-time Continuous Batch Optimization in a Semiconductor Work Center Environment

Analyse eines digitalen Zwillings zur zeitkontinuierlichen Batchoptimierung in der Halbleiterfertigung

Holger Brandl, Philipp Roßbach
SYSTEMA Systementwicklung Dipl.-Inf. Manfred Austen GmbH, Dresden (Germany), holger.brandl@systema.com, philipp.rossbach@systema.com

Hajo Terbrack
Technische Universität Dresden, Chair of Production Economy and Information Technology, Zittau (Germany), hajo.terbrack@mailbox.tu-dresden.de

Abstract: Motivated by the ongoing growth in demand for microchips, high production costs and the complex interplay of human, machine, material and method (4M), suppliers strive to develop more advanced production planning and control regimes for semiconductor production. Batching decisions often dramatically influence the overall performance of wafer fabs in terms of capacity utilization, due date compliance, cycle time and variability. To optimize such processes, we present an integrated testbed for batch formation optimization. Using a simulation of multiple semiconductor work centers, we explore how to optimize work in progress (WIP) flow with a continuous real-time scheduler and previously published batch formation heuristics. The proposed solver is designed to only optimize capacity-limited operations. By considering real-world operations requirements and semiconductor process specifics such as qualification criteria and re-entrance in our model, we demonstrate how to realize significant throughput gains. We explore and demonstrate the developed digital twin through a powerful BI frontend for historical analysis and real-time shop floor monitoring.

1 Introduction

Semiconductor production is one of the most complex production processes. Especially wafer fabrication represents a challenging task with flows of 300 to 2000 steps, a high degree of flow cyclicity and re-entrance, varying step process times ranging between a couple of minutes and 24 hours, high maintenance workload, complex and time-consuming qualification rules, heterogeneous machines within work centers, sequence-dependent setup times, a high product mix, as well as resulting average cycle times of up to 25 weeks. In addition, besides single wafer or lot processing, a considerable amount of semi frontend processes is performed in batches. In fact, up to one third of the operations in a wafer fab are performed on batch processing machines (Rocholl et al. 2020, Mönch et al. 2013, 2018).

The paper aims at contributing to both, research and practice, by embedding a real-time optimization engine developed to improve batching decisions and increase throughput into a semiconductor wafer production simulation model. For that, the article is structured as follows. First, key characteristics of semiconductor batch building as well as several methods for batch sizing, widely considered in research and industrial practice, are introduced. Then, we describe the use case, which we have modeled in a hybrid simulation environment linked to an optimization engine. Finally, we will discuss the results of our analysis by comparing the system performance against classical baseline planning and batch formation methods. The article concludes with a discussion pointing out possible improvements to the developed methods and model.

2 Batch Processing in Wafer Fabs

Lot batching is typically distinguished into s-batching and p-batching. In the former case, the process time of the batching operation equals the sum of the process times of all lots in a batch because lots are processed sequentially. However, more important in semiconductor production is the latter case, p-batching, whereby lots are processed in parallel and the batch process time equals either the maximum process time of all lots that form a batch or a specific process time irrespective of the number of lots in a batch (Mönch et al. 2009, 2011).

In contrast to other operations in semiconductor frontend production, process times of batch operations are often very long (up to 24 hours). With bottleneck management, line control is striving to maximize batch size and capacity utilization. While optimizing batch size, other production factors need to be considered as well, such as process timers, due dates, and work in progress (WIP). For instance, by off-loading multiple lots to the next operation after a batch operation is finished, long queues in front of non-batching machines and a nonlinear product flow can occur (Hopp and Spearman 2011, Mönch et al. 2011, Rocholl et al. 2020). Vice versa, line control needs to ensure that batch tools do not run dry to prevent tool standby. To form a batch, lots must meet specific process and product-dependent requirements. Typically, only lots of the same product technology can be grouped into a batch. Moreover, all lots need to be ready for processing, therefore released, placed in front of the tool and not on hold status. Furthermore, time constraints between two or more consecutive process steps may need to be considered. In that case, the associated operations have to be executed within the determined time window to avoid WIP degradation or scrap. In addition, batch-processing machines have a minimum and maximum batch size which must be ensured when processing the lots.

In summary, while determining the optimal batch size for each operation, a trade-off between high-capacity utilization on the one hand, and delay minimization, low variability and cycle time on the other hand, needs to be achieved. For that, batching decisions must be linked to the overall planning approach in a semiconductor production context, e.g., addressing tool capabilities, reacting to machine breakdowns and considering operator availability.

So far, research has addressed semiconductor batch sizing in multiple ways as in terms of queuing and control theory, look-ahead policies and several scheduling approaches (Fowler and Mönch 2022). By means of computational performance but also for the sake of understandability and implementation effort, typically, simple heuristics are used as dispatching schemes for production execution in wafer fabs. The same holds true for batching decisions as most wafer-fabs perform batch sizing by policies based on a minimum batch size (MBS) (Koo and Ruiz 2020). In such threshold schemes, a specific minimum batch size S is considered and a batch is built as soon as at least S lots which are capable of being batched together are waiting in the queue. With respect to the minimum batch size S, different batching decisions can be derived. While a value of 1 for S allows single lot processing, S equal to the capacity of the batch processing machine ensures that only full batches are loaded (Solomon et al. 2002).

More advanced batching policies were proposed consuming upstream and/or downstream WIP to be considered in batch formation. As an upstream policy example, the look-ahead batching rule by Koo and Ruiz (2020) extends the MBS scheme to emphasize potential savings in waiting time. If more time can be saved by including an upcoming lot in a batch than the delay of the existing lots in the queue, the batch loading decision is postponed until the lot arrives at the operation. On the other hand, if the resulting overall delay of the existing lots in a queue exceeds the time savings expected with the next lot arrival and delayed batch loading, the batch is built immediately and loaded to the machine.

Despite a considerable body of research in queuing theory and operations research, batch optimization remains a challenging problem with often not-realized optimization potential in wafer fabs. Furthermore, previous work often neglected the operational requirements: Because of complex work center dynamics, schedules need to be updated frequently and users expect close-to-real-time updates. Based on these considerations, this paper proposes an improved solution approach for real-time semiconductor batching formation and optimization.

3 Bottleneck Operations in Semiconductor Production

From the theory of constraints, it is known that the throughput of any system is determined by one bottleneck under steady state assumption. However, as production requirements are constantly in flux, in particular in high-mix, low-volume production facilities, bottlenecks typically vary over time as described by Chen et al. (2006). Importantly, the number of non-bottleneck steps will almost always outnumber the possible bottleneck steps by an order of magnitude.

Figure 1 shows the typical structure of a semiconductor frontend production facility, cyclic routes with lot reentrance and possible tool dedications, as well as its work center organization. Within the scope of this paper, tools are only characterized by two attributes: First, tools either perform s/p-batching or process single lots. Second, some tools are rate-limiting bottlenecks under certain production and order conditions.

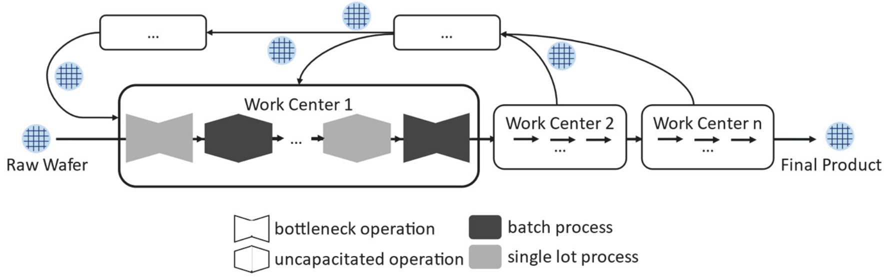

Figure 1: Semiconductor frontend production from a batching and bottleneck perspective; Importantly, the number of bottleneck operations is always much lower than the number of non-capacity limited operations within any work center. Different process flows, including some of reentrant nature, as well as different products reflect some of the complexity of the industry.

In Brandl et al. (2022), we presented a hybrid simulation model to study wafer fabrication. The factory state model proposed by Luhn et al. (2017) was used as a 4M abstraction of the semiconductor frontend production process. The model was implemented into a digital factory twin using the open-source discrete-event simulation engine "kalasim" (Brandl 2022) designed for complex industrial applications and with software development best practices in mind. The model allows studying the complex spatio-temporal alignment of production resources in the modeled frontend wafer fab. Also, complementary processes such as qualification, maintenance, engineering, logistics, and material preparation as well as operator activities were included in the model. The model allows for configurable route-definitions, s/p batching, sequence-dependent setup, and emits metrics for tools, lots and individual wafers.

4 Batch Formation Optimization

For the study at hand, we complemented our simulation model of Brandl et al. (2022) with a more configurable dispatching engine. First, we added support for configurable batching rules. Second and more importantly, we implemented support for hybrid execution planning in the following way: We propose a hybrid execution planning and optimization model by which only capacity-limited operations are scheduled globally, and non-limited operations are governed by event-driven dispatching rules. With this hybrid line-control model, the computational requirements of WIP flow optimization and batch formation across work centers are significantly reduced.

This flexible job-shop problem (FJSP) scheduler with batch formation support was implemented using a chained graph approach with tools as roots within the framework proposed by De Smet (2006). Batches are allowed to be formed based on a configurable batching criterion within a minimum and a maximum batch size. If adjacent lots along a solver chain satisfy the batching criterion but do not fulfill the minimum batch size requirements, lots are not batched. From a process perspective these lots would still be run along with test-wafer batcher, adding to the overall production costs. The objective function penalizes such runs. While the simulation model is capable of consuming cyclicality with regard to process routes, reentrant flows are not yet implemented in the embedded optimization model described next.

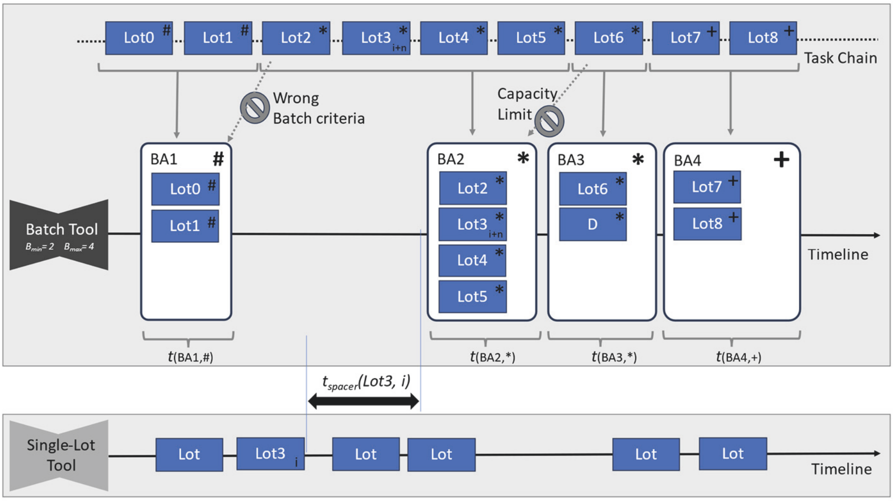

Figure 2: Batch & Timeline formation. In the example the single processing and the batch tool are considered as bottlenecks. For batch formation, sequentially aligned lots of the chained graph approach rooted in the tool, are evaluated for stretches of matching batch criteria and capacity. To compute time coordinates, additional restrictions based on the individual routes are considered. For example, a Lot3 which is first scheduled at the single-lot tool for its process step i, needs to undergo n additional steps not included the optimization process before arriving at step i+n. This is modelled via a spacer element t_{spacer}(Lot3, i) to ensure feasibility of the computed execution plan. Different batches will have varying times t(BA) depending on the operation-dependent tool recipes. Because batch formation is independent from scoring, some solutions would lead to additional costs, such as Lot6 which is not properly batched with WIP and thus is processed along with a dummy wafer cassette D.

The solver (Figure 2) is implemented as a 2-step process: First, the schedule structure by evaluating batching criteria discussed above to form batches in a greedy manner starting from the origin of the tool chain. Secondly, with batches in place, the solver computes the final schedule by considering virtual spacer times between process steps. These process step dependent spacer elements along each individual lot timeline are set to the average cycle time between solver-optimized bottleneck processes.

Multiple constraints of the proposed bottleneck scheduler were incorporated to support optimization using different score levels indicating severity of the score and to allow for balancing of metrics within the objective function of the scheduler. Non-maximal batches are penalized directly, the lack of batch formation (e.g., single lots) is penalized because unbatched WIP is typically either stalled or batched with dummy wafers (depending on the process) leading to higher production costs. The solver model accounts for sequence-dependent setup, tool qualification settings (counterbased requalification), operator availability, due date compliance, and lot prioritization. Because of the complex business domain, we believe a constraint programming approach with configurable score weights to be a more efficient problem formulation than other more classical representations such as mixed-integer programming.

5 Results

As a baseline for production execution and batch building, we include several dispatching rules commonly used in industrial practice. For conventional dispatching schemes which do not take into account batch building by default, we consider a "greedy approach" by which a batch is formed if more than one lot of the same operation and product is available for processing at the tool in the moment the next lot is dispatched. In addition, we extended our dispatching engine to support previously published batch formation heuristics. When these batching heuristics are executed, equipment not capable of batch processing are ruled by conventional dispatching schemes instead. Furthermore, we adjusted the batching heuristics to allow for batching with a smaller batch size than the minimum as soon as lots were waiting more than 24 hours in front of an operation.

In addition to the proposed hybrid bottleneck scheduler, we considered the following baseline heuristics for characterizing production in our model:

- First In First Out scheme with greedy batch building
- Earliest Due Date scheme with greedy batch building
- Setup Avoidance scheme with greedy batch building
- Minimum Batch Size rule with different values for minimum batch size
- Look Ahead heuristic with different values for minimum batch size

To study the model, we configured a simulation scenario with two adjacent work centers, namely epitaxy and furnace. Each work center is modeled with a single bottleneck operation. All products shared the same route from epitaxy to furnace, but were configured with different tool recipe parameters. Each route was configured with four additional non-capacity limited steps. For conducting experiments with our model, we parameterized the simulation in accordance with an existing frontend production of one of our manufacturing partners. The furnace step was modeled as p-batching tools with a minimum and maximum batch size of 2 and 4, respectively. WIP was released stochastically using a non-uniform distribution over the 20 products. Due dates were stochastically modeled around the average cycle time of 4 days with a variance of 1 day. The used batching criteria was set to only batch lots of the same product at the same operation. One measurement operation along the route occurred twice, to reflect typical route topologies in semiconductor frontend production.

All simulation experiments were performed under controlled randomization conditions with replication except for the real-time integration experiments (see chapter 6). Both the simulation and the used multi-threaded bottleneck scheduler were configured and tested to provide deterministic results. Results and metrics were computed, analyzed and visualized in R.

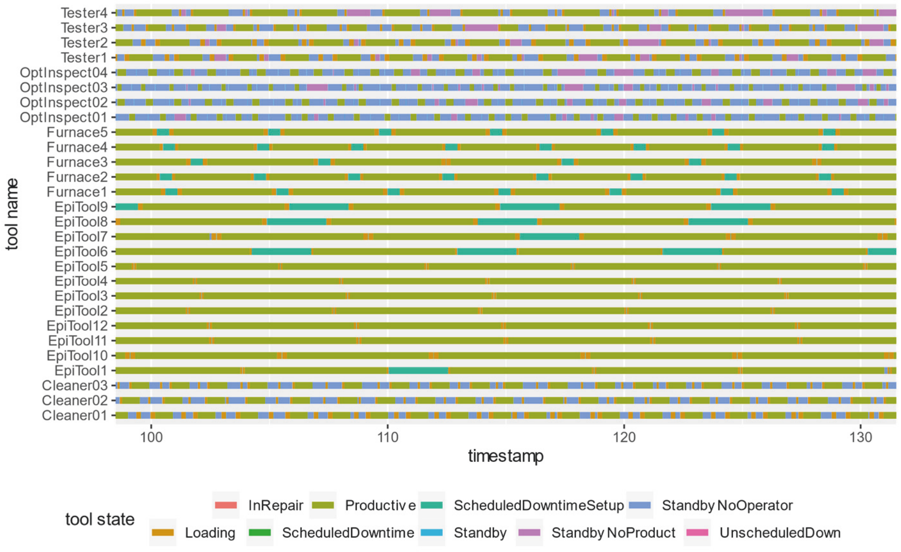

Figure 3: Example production schedule when using Look-Ahead heuristic with a minimum batch size of 3 for batch formation. Indicated by color are the various tool statuses. Depending on the work center, there are yet different challenges, from considerable setup efforts at the epitaxy reactors, to understaffing at the optical inspection.

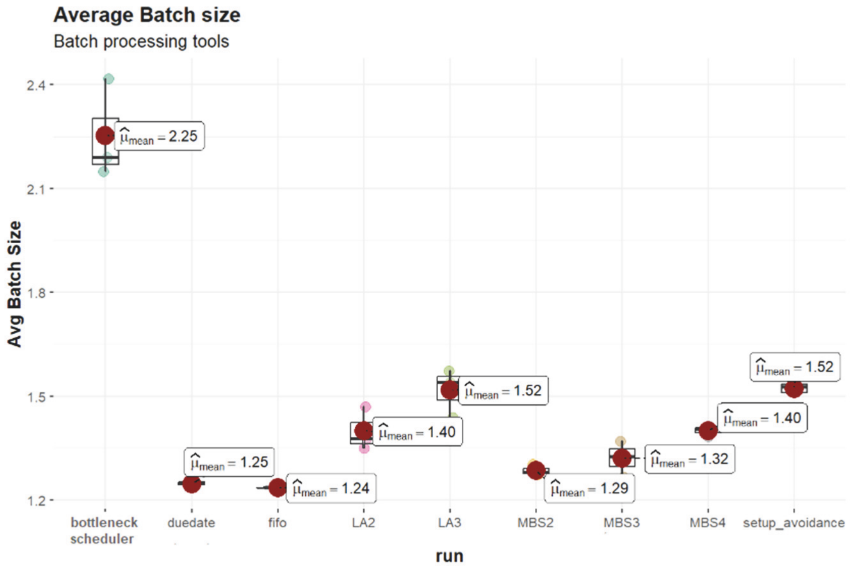

Figure 4: Average batch size along simulation runs given for each considered heuristic. The proposed bottleneck scheduler was found to outperform other methods regarding the average realized batch size in the resulting schedule averaged over multiple runs.

To account for different production objectives, we computed different process KPIs which are compared in Figure 5. To enable industrial engineers, KPIs were normalized into a radar chart display. Intuitively, the object is to maximize the area of the polygon.

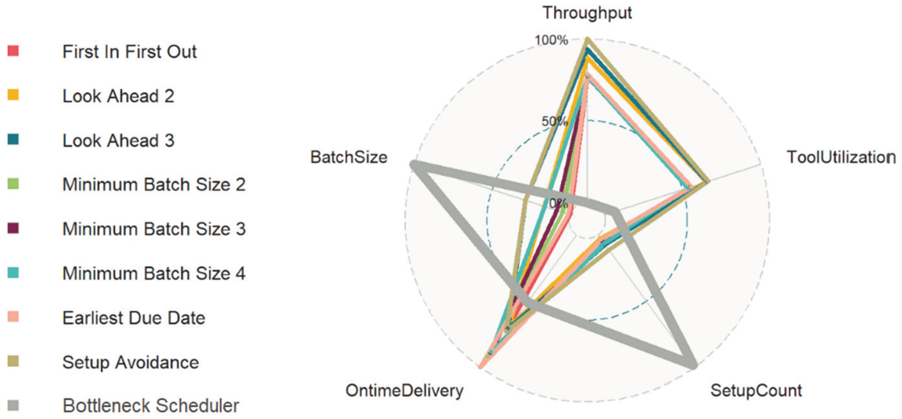

Figure 5: Due to multiple objectives relevant in semiconductor production planning, resulting production plans were analyzed in terms of several KPIs and normalized to allow for comparison. In its current configuration, the bottleneck scheduler is highly optimized with regard to batch sizing but sacrifices a lot tool utilization. This common challenge requires a careful balancing of partially conflicting production objectives based on business and operational consideration.

6 Real-time Fab Optimization

In most frontend operations, scheduling is still an underused optimization method due to the challenging combinatorics, high production dynamics and computation requirements. Tool assignments are often only indirectly computed via ordering schemes that sort WIP given lot requirements and tool capabilities. The advantage of doing so is the ability to dispatch material in real-time by consuming factory changes via a message-oriented middleware.

Still, traditionally a solver would still require a recompute model, where the schedule is temporarily out-of-sync with the shopfloor while being recomputed. To overcome this limitation, we have operationalized the proposed bottleneck scheduler using a real-time change model proposed by De Smet (2006): Material changes are propagated into a running solver using a framework provided change API. Instantaneously, the solver picks up these changes when exploring a better solution. With this model, scheduling can be realized as an event-driven planning function similar to traditional lot dispatching. Another benefit of the proposed model is that the user is no longer required to define termination conditions, which may miss the most optimal solution.

7 Discussion

A major challenge of scheduling in semiconductor production is the high process complexity, its high dynamics, and the resulting invalidation of an existing schedule. Schedule recomputation is severely limited by CPU time, in particular when optimization WIP flow and batch formation across work centers. Our proposed hybrid batch formation and route optimization model aims to combine both, scheduling and dispatching to maximize batch formation. By complementing a planning heuristic for capacity limited process steps with an event-driven real-time dispatcher, we propose a best-of-breed fusion planner to optimize workload across a series of work centers. Following this approach, we consider several work centers with bottleneck operations and batch processing machines that are controlled by a scheduling engine while the remaining process steps along the process flows are ruled by a dispatching module.

During the simulation study, multiple aspects were found to have potential for further investigation and improvement. First, WIP flow is currently enforced via constraints, leading to much more complex combinatorics. By using built-in hard constraints when exploring the solution space in the solver, we plan to further improve solver performance considerably. Also, by design, the bottleneck scheduler was found to be not yet suited to form batches in reentrant flows. As individual moves/changes in a FJSP require widespread adjustments to temporal and structural properties, it remains to be shown if the design can be extended for reentrant flows as well.

By running the scheduler as an event-driven real-time planning engine next to the shopfloor simulation model, different batch heuristics and factory KPIs could be effectively studied in an integrated testbed. Because of the used code-first and not UI-centered simulation framework, we could enable multiple engineers and analysts to evolve the model collaboratively. For an extended analysis and additional metrics, the reader is referred to a supplementary jupyter notebook (https://www.systema.com/asim2023).

References

Brandl, H., Rossbach, P., Terbrack, H., Sprogies, T. (2022): Maximizing Throughput, Due Date Compliance and Other Partially Conflicting Objectives Using Multifactorial AI-powered Optimization, Winter Simulation Conference 2022, December 11-14, Singapore.

Brandl, H. (2022). “Code-first Process Modeling and Analysis with Kalasim”. FOSDEM’22, https://fosdem.org, accessed 8th February 2023.

Chen K.S., Lee, M. S., Pulat, S. P., A. Moses, S. (2006): The shifting bottleneck procedure for job-shops with parallel machines, International Journal of Industrial and Systems Engineering (IJISE), Vol. 1, No. 1/2, 2006.

De Smet, G. (2006). “OptaPlanner User Guide”, Red Hat Inc., https://www.optaplanner.org, accessed 8th February 2023.

Fowler, J. W., Mönch, L. (2022). A survey of scheduling with parallel batch (p-batch) processing. European journal of operational research, 298(1), 1-24.

Hopp, W. J., Spearman, M. L. (2011): Factory Physics, 3nd edn. Waveland Press.

Koo, P. H., Ruiz, R. (2020). Simulation-Based Analysis on Operational Control of Batch Processors in Wafer Fabrication. Applied Sciences, 10(17), 5936.

Luhn, G., Ertelt, M., Zinner, M. (2017): Real-Time Information Systems and Methodology based on Continuous Homomorphic Processing in Linear Information Spaces; EP 3114620 A1,

Mönch, L., Fowler, J. W., Dauzère-Pérès, S., Mason, S. J., & Rose, O. (2009). Scheduling semiconductor manufacturing operations: Problems, solution techniques, and future challenges. In: 4th Multidisciplinary International Conference on Scheduling: Theory & Applications, Dublin, Ireland.

Mönch, L., Fowler, J. W., Dauzère-Pérès, S., Mason, S. J., & Rose, O. (2011). A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. Journal of scheduling, 14, 583-599.

Mönch L., Fowler J. W., Mason S. J. (2013): Production Planning and Control for Wafer Fabrication facilities: Modeling, analysis, and systems. Springer, New York.

Mönch, L., Uzsoy, R., Fowler, J. W. (2018). A survey of semiconductor supply chain models part I: Semiconductor supply chains, strategic network design, and supply chain simulation. International Journal of Production Research, 56(13), 4524-4545.

Rocholl, J., Mönch, L., & Fowler, J. (2020). Bi-criteria parallel batch machine scheduling to minimize total weighted tardiness and electricity cost. Journal of Business Economics, 90(9), 1345-1381.

Solomon, L., Fowler, J. W., Pfund, M., Jensen, P. H. (2002). The inclusion of future arrivals and downstream setups into wafer fabrication batch processing decisions. Journal of Electronics Manufacturing, 11(02), 149-159.