MULTI-OBJECTIVE PARALLEL BATCH SCHEDULING IN WAFER FABS WITH JOB TIMELINK CONSTRAINTS

Semya Elaoud
Ruaridh Williamson
Begun Efeoglu Sanli
Dennis Xenos

Flexciton Ltd
145 City Road
London, EC1V 1AZ, UNITED KINGDOM

ABSTRACT

We consider multi-objective batch scheduling in the complex flexible job shop problem applied to semiconductor wafer fabs. Batches have different operating costs and consecutive steps of a job are constrained with timelinks. We also consider several other process aspects that arise in semiconductor wafer fabrication facilities such as flexible machine downtime, incompatible job families, different job sizes and parallel machines. The aim is to minimize the total weighted batching cost, queuing time, and the number of violations of timelink constraints. We present a hybrid two-stage solution strategy, combining Mixed Integer Linear Programming (MILP) models and heuristics. At a high level, the proposed approach can be broken down into "constructive" and "improvement" steps. The comparison of Flexciton schedules evaluated under uncertainty against factory schedules when solving large industrial instances shows the significant improvements that our solution can bring.

1 INTRODUCTION

The world is currently witnessing a global semiconductor chip shortage; a crisis so considerable that the world's largest tech companies are suspending the launch of their products (Sweney (2021)). This shortage is a result of many factors including an increasing demand, lack of raw materials, and the limited production capacity of semiconductor facilities. According to the Semiconductor Industry Association, global semiconductor sales increased 6.5% to $439 billion in 2020 according to SIA (2021), showing a significantly growing demand for chips from automakers and tech giants. Multi-billion dollar wafer fabrication plants run round-the-clock. Yet the semiconductor manufacturing is a very capital intensive industry; adding significant new wafer fabrication capacity can take years and billions of dollars. For most semiconductor companies, one of the best ways to capitalize on this surge in demand is to cultivate efficient production processes in order to decrease production costs and increase the time-to-market ratio. Efficient production control and advanced scheduling are vital in modern wafer fabs as a method to increase throughput, reduce queuing time, and reduce rework and scrap.

In order to improve the high level Key Performance Indicators (KPIs) of a fab, many state-of-the art schedulers and dispatching algorithms will focus on scheduling the Work-In-Progress (WIP) either immediately in transit to the toolset or already on the rack at the toolset. Such an approach allows for these techniques to scale across large numbers of toolsets easily as they don't need to consider the knock-on impact of their localised scheduling decisions. That said, improving upon this "local scheduling" of toolsets can still realise significant gains for a fab where the decisions made at each toolset are independent. In Kopanos et al. (2020) Flexciton reported a significant reduction of more than 43% in cycle times for high-

priority wafers, a reduction of about 9% in total cycle times, and a 7% increase in throughput; compared to the simulation scheduler for the metrology toolsets at Seagate's Springtown facility. Having shown improved efficiency at a toolset level where each toolset is optimized separately, in this paper we outline the efficiency of Flexciton's solution at a global level, taking into consideration different schedulable toolsets simultaneously. We present a hybrid two-stage solution strategy based on job decomposition comprised of an MILP solution and local improvement loop. This solution is used for solving the global scheduling problem; also known as the general scheduling problem with time constraints.

2 PROBLEM DESCRIPTION

Wafer fabrication is one of the most complicated manufacturing processes in the modern world. A single lot of wafers may go through over 1,000 steps in different work areas resulting in complex constraints and major dependencies.

2.1 Timelink Constraints

Timelink constraints (also known as time lags in scheduling literature) occur when a set of consecutive process steps must be completed within a fixed time window. In a highly complex wafer fabrication environment, these constraints add significant complexity; even the most advanced fabs struggle with scheduling time constraints.

A silicon wafer undergoes a fabrication process by entering multiple production steps, where each step is performed by different, highly sophisticated tools. Optimizing the transition and waiting time of the lots has a huge impact not only on a fab production performance but also on its profitability. As an example, by introducing time constraints at the wet etch and furnace process steps, we prevent the likelihood of oxidation and contamination. Failing to do so risks contact failures, low and unstable yields, the consequence of which is either rework or the wafers must be scrapped. Such problems are difficult to discover during wafer processing, and to run special monitoring lots would be a considerable effort.

Yield optimization has long been considered to be one of the key goals, yet difficult to achieve in semiconductor wafer fab operations. As the semiconductor manufacturing industry becomes more competitive, effective yield management is a determining factor to deal with increasing cost pressures. Timelinks between consecutive process steps are one of the most difficult constraints to schedule, with a significant impact on yield management.

Some factories avoid the problem by dedicating tools to each process group that requires a previous cleaning or etch step. The disadvantage of this strategy is the higher demand from the wet tools, which leads to higher investment, more cleanroom space, and ultimately to lower capital efficiency. The tradeoff between increasing throughput and a higher likelihood of violating lots' time constraints is an everyday battle for fab managers trying to meet yield targets.

An example of time constraints for a single lot is illustrated in Figure 1. It shows a timelink system between four consecutive process steps. In this example, we can see that the lot has timelinks constraining Step 2 to Step 4 as well as from Step 3 to Step 4, with overlapping timelink phases. This means that after completing process Step 3, the lot begins a new timelink phase (Time Link 3) whilst already transitioning through an existing timelink (Time Link 2) started upon completion of Step 2. As you might expect, the need to simultaneously look ahead and consider future decisions whilst also being constrained by past decisions is not trivial to model well in a heuristic or as dispatch rules.

Timelink constraints are already difficult to navigate, but nesting them adds yet another layer of complexity for heuristics. Different types of timelink constraints are analyzed and discussed in Klemmt and Mönch (2012). In Figure 1, if the final Step 4 cannot be brought forward, scheduling Step 3 too close to Step 2 may make it impossible to meet the "Time Link 3". This is because the time between Steps 3 and 4 is now greater than the maximum allowed. This would not be a problem if the timelink constraints were not nested and we only have to schedule according to the "Time Link 2".
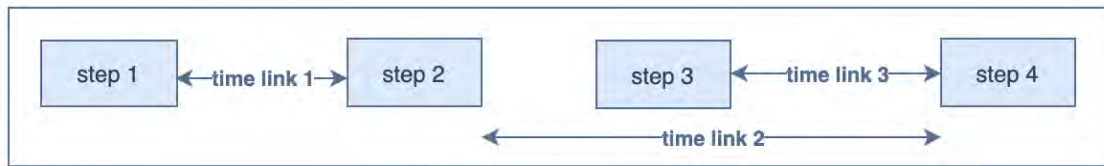
Figure 1: An example of timelink constraints for a single lot.

2.2 Global Scheduling with Timelink Constraints

Timelink constraints endorse the necessity for a global fab scheduler. These production constraints mean that toolsets become tightly coupled and must be optimized as a single entity. Without doing so, the WIP flow cannot be controlled well and we may end up creating bottlenecks at downstream tools, thus resulting in timelink violations due to a queue build-up.

For example, maximising throughput at the upstream toolset may cause too much WIP to arrive at the downstream toolset to process before the timelink expires. These lots end up queuing in front of the downstream toolset and ultimately violate their timelink due to waiting too long before processing. It is therefore common to see large queuing times of lots at the toolsets where timelink constraints commence and very little queue time in front of the remainder of the downstream toolsets.

On one hand, the WIP needs to flow freely through downstream toolsets with minimal queuing however on the other hand, we cannot afford to be overly conservative and stifle throughput as a result of keeping downstream tools unnecessarily idle waiting for transiting lots. This problem enforces the need for so-called "global" scheduling of the fab that considers upstream and downstream toolsets simultaneously.

Global fab scheduling problem can be modelled as a complex flexible job-shop-scheduling problem with timelink constraints. A flexible job-shop-scheduling problem is an extension of classical job-shop problems that permit an operation of each job to be processed by more than one machine Chan et al. (2006). The most common solution approaches can be found in the literature are integer programming and metaheuristics. A simple heuristic that sequentially schedules the jobs in a list scheduling manner and a decomposition approach based on mixed integer programming are proposed in Klemmt and Mönch (2012). The problem was simplified and reduced to a flexible flow shop scheduling problem. The authors in Yu et al. (2013) presented a two-stage lot scheduling mixed integer programming (MIP) approach with time constraints for small problems and an efficient solution procedure for larger problems. This procedure decomposes the jobs into two subsets, whereby only one subset of jobs is scheduled by the MIP and the other one scheduled with dispatching. The suggested MIP model solutions problems having up to 25 lots. Population based metaheuristic are also frequently found in the form of genetic algorithms and tend to provide acceptable results. A two-stage genetic algorithm for a flexible job-shop scheduling problem with sequence dependent attached/detached setup, machine release date and lag-time is developed in Defersha and Rooyani (2020). A simpler list-based scheduling heuristic from Sadeghi et al. (2015) and Klemmt and Mönch (2012) offered an interesting compromise between solution quality and scalability and did not limit the way time constraints are modeled. Knopp et al. (2017) developed an algorithm applying Simulated Annealing embedded in a Greedy Randomized Adaptive Search Procedure (GRASP) and provided a good approximated solution in short execution times. Nattaf et al. (2019) proposed an integer linear programming model and a constraint programming model, as well as two improvement procedures of existing two heuristics. For the sake of simplicity, most of these works do not consider the important operational sides of the problem; batch scheduling, job incompatibilities and/or machine downtimes are often ignored.

2.3 Multi-Objective Optimization

Scheduling of semiconductor wafer fabrication is a complex system usually involving multiple and conflicting objectives. Although multi-objective scheduling problems have been widely studied in the literature, a

minor number of papers considered multi-objective scheduling in semiconductor industry. Baez Senties et al. (2006) and Senties et al. (2010) proposed an artificial neural network based-technique embedded into a multi-objective genetic algorithm for a multi-decision scheduling problem in a semiconductor wafer fabrication environment. In a recent work Tamssaouet et al. (2021), the authors introduced two a priori multi-objective extensions of Simulated Annealing using a combination of a lexicographic order and weights embedded in a GRASP based approach. In their preference model, trade-off is only allowed between some criteria.

In this paper we consider the optimization of three of the most important objectives in semiconductor industry:

- Minimize the number of violations to timelink constraints: as pointed out in section 2.1, violating those constraints might result in rework, scrap or longer cycle time. In some circumstances, due to limited capacity a minimal number of violations may be unavoidable. Therefore, they are treated as "soft" constraints; allowed to be violated and modelled by heavily penalising their violation in the objective function.
- Maximize the batching efficiency weighted by tool. This is to model the relative cost to the fab of running batches on certain tools over others. It also allows the optimizer to distinguish worsening the batching efficiency at upstream toolsets to realise even greater gains at the more expensive downstream toolsets.
- Minimize the queuing time of lots. This naturally translates to a reduction in cycle time which is often a fab's key objective.

The number of late lots deliveries is not optimized as an objective. Instead, a maximum number of late lots is treated as hard constraint such that their number must be no worse than the baseline scenario in all cases.

2.4 Weighted Number of Batches

In semiconductor manufacturing, the effective monetary cost of running a batch of lots can vary greatly across different tool groups. Therefore the creation of batches is penalized according to a hierarchy of three costs:

- Clean tool (batches are inexpensive)
- Cheap furnace tool (batches are eg. x5 more costly)
- Expensive furnace tool (batches are eg. x10 more costly)

To the best of our knowledge, batch scheduling with different batch costs and job timelink constraints in a multi-objective context has never been addressed in the literature. Furthermore, it seems that most of the academically motivated papers do not consider different batching costs and mostly focus on minimizing makespan or number of late lots. In this paper, we address the optimization of three of the most important KPIs in semiconductor manufacturing while considering complex process conditions found in wafers fabrication facilities.

3 SOLUTION STRATEGY

The Flexciton solution strategy for solving global scheduling problems consists of a hybrid of MILP models and heuristics. At a high level this strategy can be broken down into two constructive and improvement stages. The constructive step produces a high-quality schedule quickly, typically within 2-3 minutes. The improvement step refines this schedule carefully, leading to a better solution for a further 2-3 minutes. These two steps are encapsulated within a distributed processing environment of multiple threads processing tasks in parallel. The modelled process is showed in Figure 2.
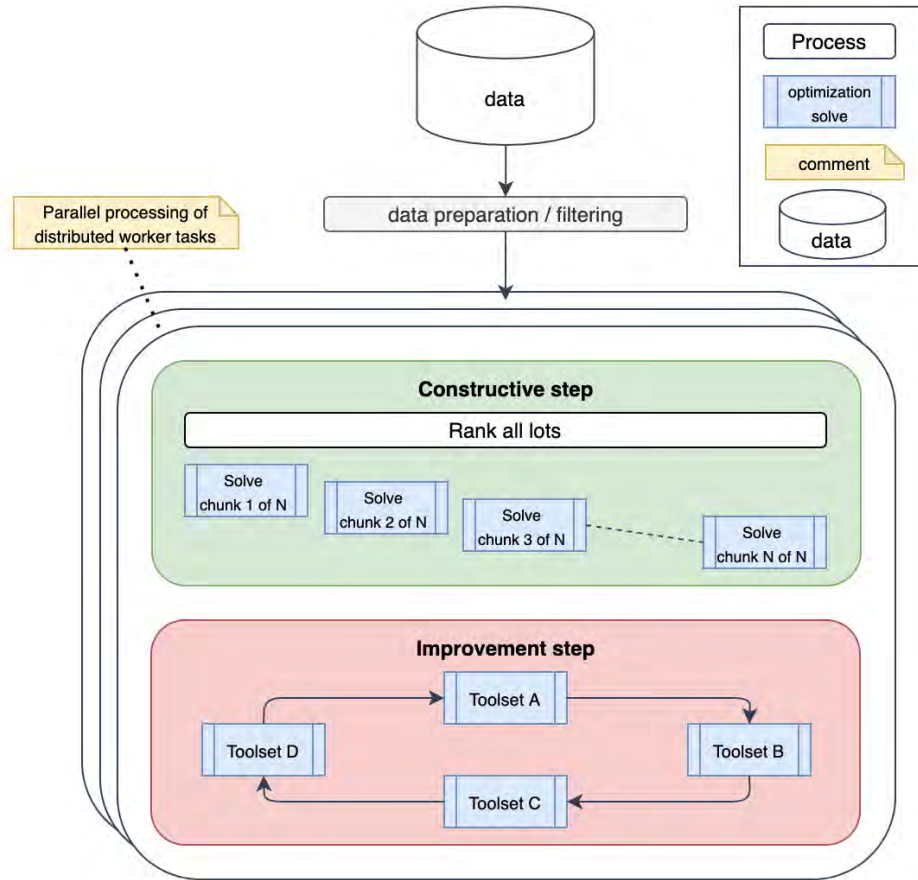
Figure 2: Summary of Global Scheduling solution strategy.

3.1 Constructive Step

The constructive step is focused around an iterative process of adding decreasingly important lots into the schedule. All the lots are ranked according to a combination of criteria, φ(lot). They are then scheduled in N subsets, a predefined number of iterations. Higher priority is allowed to jobs with earliest due date, more timelink constraints and higher number of steps to be scheduled. This constructive step is a modified version of the extension of the list scheduling procedure Klemmt and Mönch (2012) where jobs are sorted in non-decreasing order with respect to their due dates. Adding the number of timelink constraints and the number of steps to schedule gives a better view on the priority of a job. Increasing the priority of these lots results in scheduling them in earlier iterations where there is more freedom in the schedule. We also ensure to consider all future and timelinked steps of a lot simultaneously in the iteration for which it is selected.

In taking a job decomposition approach, we iteratively solve smaller subproblems where each subproblem contains only a subsets of the total number of jobs. This reduces the total complexity of the problem, which is principally derived by the number of jobs to be scheduled. Subproblems are iteratively solved using MILP.

Inspired by the efficiency of the list scheduling procedure Klemmt and Mönch (2012), the algorithm of this constructive step can be described as follows:

1. Determine the bottleneck toolsets coupled by timelink constraints
2. Build an MILP model encompassing these coupled toolsets

3. Rank all lots according to φ(lot) in descending order
4. Split the ranked jobs into N chunks of equal number of schedulable steps
5. Solve the MILP problem of each iteration, adding each chunk of jobs in successive iterations

During the iterative solutions of the MILP, jobs of previous iterations are not allowed to move batches however they may still change their timing and/or sequencing. The maximum running time of this stage is evaluated according to the number of jobs to be scheduled. The division of the total time limit is skewed towards earlier iterations as these are typically more complex subproblems to solve due to the ranking criteria of lots. The advantage of this step is the possibility of including all timelink constraints of jobs in each chunk. Otherwise, solving the full problem with coupled timelink constraints for industrial size datasets would not be possible in a reasonable amount of time.

3.2 Improvement Step

The second and final phase of the solution strategy consists of cycling through the bottleneck toolsets considering one toolset at a time. Given a full MILP model of the entire fab, we only allow decision variables related to the toolset at hand to be modified by the solver in any given iteration. In doing so, the problem size is effectively reduced however we ensure the impact on the timing of the steps on other toolsets is also still considered in the objective function of this toolset.

Furthermore, we allow lots to move within a "close neighbourhood" of candidate batches. This neighbourhood is defined for a given batch B as any other batch B^* that has been pre-assigned the same recipe and starts or ends within α minutes of the candidate batch. To simplify this search, batches that are already full on toolsets with a high maximum batch size are considered immutable; their lots are excluded from this local neighbourhood search. We also allow the search to move lots between tools of a toolset. This improvement step procedure continues iterating until either all toolsets have successfully returned an optimal solution in a single pass or the predefined time limit has been exceeded.

3.3 Lexicographic Objective Function

Another facet of this solution strategy is the objective function that is used for the MILP model. A linear weighted sum of different objectives is typically how objective functions with many different KPIs are composed. In this instance, however, the objectives have significantly different priorities and having linear weights that are several orders of magnitude different can cause the optimizer to become unstable.

For this reason, we elect to change the objective function to be hierarchical; optimize first objective $f_1(x)$ only, then optimize the second objective $f_2(x)$ such that $f_1(x)$ does not worsen by $\mu\%$ (typically 0-10%). This is especially useful when dealing with different orders of magnitude in the objective terms of considerably different coefficients because only one subproblem is considered at a time. This further enables us to model the timelink constraints via an objective function penalty term and once the number of timelinks has been minimized without considering any confounding KPIs, the remaining terms can be optimized given that number of timelink violations. Optimizing timelinks is best suited to a lexicographic objective for three reasons:

1. *Infeasibility*: Should timelinks be treated as hard constraints, the problem may end up being infeasible due to potential limited fab capacity. Hence we minimize them as part of the objective function instead of disallowing any violations at all.
2. *Importance*: Additionally, a lexicographic objective gives us the ability to highly prioritise this objective term above all else. Since the cost of violating a timelink is so great to a fab (due to potentially scrapping a wafer or performing rework), any improvements in the other KPIs are overshadowed by this term.
3. *Automation*: The Flexciton solution is deployed in a live application where the time available for decision making and human interventions is limited, therefore relevant objective preferences should

be set up-front and optimized accordingly.

We therefore setup the objective function with the following ranking as outlined in 2.3:

1. Above all else, eliminate all timelink violations as much as possible
2. Secondarily, maximise the batching efficiency weighted by tool
3. Finally, minimize the queuing time of lots

Notably, solving the problem in this fashion could lead to a situation where the queueing time of lots suffers at the expensive of optimal batching efficiency and timelink violations; the solution is not the same as if all terms were optimised together. In using such an approach, we have also considered that the objectives are in order of "freedom". That is to say that when optimising the third term, a constrained number of timelink violations and batching efficiency still allows a large number of solutions to be explored. We have not constrained precise jobs in the way that optimising queueing time (or say, job end times) *first* would do. That said, there is certainly scope for future work in this area of fine-tuning the lexicographic order of our objective function.

3.4 Parallel Computing

As the modelling parameters need to be decided a priori, we utilise parallel computing to launch similar computations with subtly different configurations and inspect the results upon completion. Typically we run approximately 10 different parallel threads whose output schedules are assessed according to a success criterion chosen by the parent thread.

4   BENCHMARK STUDIES

To benchmark the scheduler, we focus on the optimal multistage scheduling of single-recipe batching machines with timelinks. In order to appropriately consider the timelinks, we also model the toolset(s) triggering the start of the timelink constraints. An example of such an area is the diffusion (aka furnace) and clean toolsets of the semiconductor fab.

In order to accurately model the upstream toolsets, we must also consider their WIP that is *not* subsequently entering a timelink. To exclude this WIP would overestimate the capacity of that toolset to schedule timelink WIP. These lots are modelled via a pre-specified delivery date prior to which they must reach their downstream toolset. This ensures that the lots are scheduled in a timely manner without explicitly modelling their downstream toolset(s).

4.1 Assumptions

- All scheduling data are assumed to be deterministic. The scheduler estimates uncertainties by modelling various distributions and suitable values are sampled from these distributions before optimizing.
- Multiple processing steps are considered per available wafer and all of these are known with certainty. There are no conditional steps.
- Batch loading and unloading times do not depend on recipes or wafers.
- No preemption is allowed, i.e., once a batch starts, it cannot be interrupted.

4.2 Results

To assess the performance of the proposed approach, we consider ten benchmark datasets of varying levels of throughput (ie lot moves), and online machines such that each dataset represents a different fab state. Descriptive metrics of the benchmark datasets are given in Table 1. The second column reports the throughput level for each data set, ranging from 59 to 704 wafer steps. Since not every lot-step can be

Table 1: Dataset descriptive metrics.

Dataset	Num. wafer steps (throughput)	Avg. candidate lot steps per machine	Num. pairs of steps linked with time link constraints
1	59	6.82	5
2	77	9.00	5
3	135	12.21	1
4	196	24.11	40
5	483	35.33	97
6	520	39.11	135
7	589	45.46	156
8	630	45.96	195
9	627	46.80	164
10	704	51.11	191

processed on every machine, the average number of eligible lot-steps per machine is shown in the third column. Although a reduced number of valid lot/machines combinations results in a reduced number of binaries in the MILP model, it also reduces the potential to improve schedule quality. The last column reports the number of pairs of steps linked with timelink constraints. The number of machines ranges from 50 to 70 for all data sets.

The baseline case results are created using simulation techniques involving heuristic rules derived from the current state-of-the-art in scheduling this toolset group. These heuristic rules are akin to the standard practice of real-time dispatch systems in a live fab. For the timelink toolset group in question, this involves a pull approach for order release whereby when an upstream toolset is idle or near-idle, it requests lots to be dispatched at the upstream toolsets. For details of such an approach see Mönch et al. (2013). This is a common state-of-the-art approach for dispatching algorithms to deal with timelink constraints albeit inefficiently.

The optimized case must schedule the same number of moves as the baseline case; throughput must be the same for both cases in each dataset.

In Table 2, we report results comparing the percentage change in the three optimized objectives between the baseline schedule and the optimized schedule. The third column of the table shows the improvement percentage of the first objective in the lexicographic order as stated in §3.3. Datasets with -100% violated timelink change means that the proposed approach managed to provide a solution with zero timelink violations, i.e. all violations that were present in the baseline schedules were removed in the optimised version. On average, the proposed solution eliminated 72.2% of timelink violations across all datasets. A lower level of detail is given for the batching efficiency objective (columns 4-6) where percentage improvement in batch throughput is given for each of the various cost levels. The weighted batching objective can easily be derived by aggregating different batching improvement percentages. The last column shows the change in queuing time for all wafers. A negative percentage indicates a lower total queuing time for the proposed solution. Overall, results in Table 2 show that the baseline solution is dominated by the optimized schedule in all objectives simultaneously in almost all datasets. Finally, the number of late lot deliveries is not reported in the table as it was zero for all baseline cases except for Dataset 7 where it was 2 lots. The optimized case saw no change to these results.

In general, as the throughput increases, the opportunity to improve batching increases due to simply more batches being present overall. The expensive batching column in Table 2 present the lowest improvement rates as the number of expensive batches is originally low across all datasets. In general there is higher utilization of moderate and cheap batching toolsets which explains the higher average improvement rates compared to the most expensive batches and tools. This also accounts for the unusually high queuing time

Table 2: Solution performance (percentage change compared to the baseline schedule).

Dataset	Num. wafer steps (throughput)	Violated timelinks % change	Num. batches % change			Queuing time % change
			Expensive	*Moderate*	*Cheap*	
1	59	0.0	Nil batches	0.00	-18.2	-89.7
2	77	0.0	0.0	0.00	-23.1	-79.5
3	135	0.0	0.0	-33.3	-16.7	-48.7
4	196	0.0	0.0	-18.2	-35.7	-59.0
5	483	0.0	4.8	-16.9	-26.2	-36.1
6	520	-100.0	0.0	-15.9	-18.3	-48.9
7	589	-100.0	-8.7	-18.2	-6.6	-30.6
8	627	-100.0	-4.3	-21.9	-19.3	-17.5
9	630	66.7	0.0	-11.4	-10.6	-16.5
10	704	0.0	-3.3	-15.2	-11.6	-35.3
Average		-72.2	-2.0	-17.0	-16.2	-39.0

reductions in the low WIP scenarios. It would be typical of heuristic rules that wait for minimum batch sizes to cause unnecessarily high queuing times in low WIP conditions.

Results in Table 2 show comparable improvement rates for all data sets, the real cost impact is, however, more important for higher WIP situations. Despite the number of batches being the primary objective in this case study, we find that the optimizer still manages to significantly reduce queuing time in addition. This is a notable finding for optimization techniques; for the same level of batching, queue time can be decreased without any additional tuning of the solution. This is because the lexicographic approach offers the ability to the optimizer to assess that if no batching gains can be achieved it focuses on the next most important KPI without worsening the former objective.
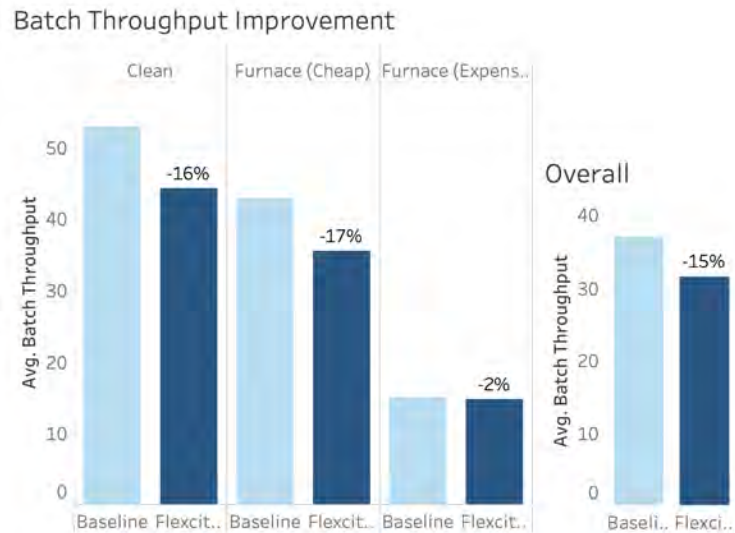

Figure 3: Batch Throughput Improvement.

Figure 3 summarizes the average batch throughput improvement rates broken down by different batching cost levels. It also shows a significant overall batching improvement across all toolsets. Notably, 15% of all batches were saved. Since the cost of the batch was dictated by the toolset, it was notably harder to cut the expensive batches as fewer lot steps were eligible for these toolsets.
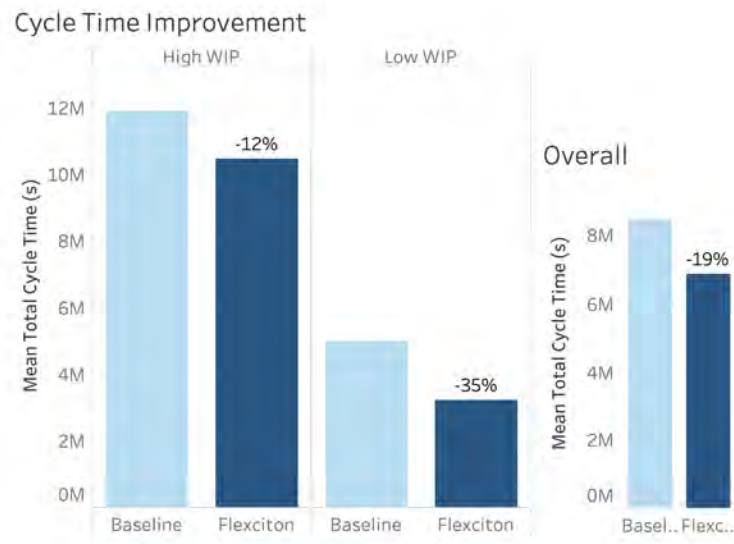
Figure 4: Cycle Time Improvement.

Figure 4 shows the the cycle time improvement split by the five highest throughput datasets against the lowest five. Overall the solution saves 19% of cycle time compared to the baseline scenario with greater gains realised in the low WIP cases.

5 CONCLUSION

In this paper we addressed the complex problem of batch scheduling with varying batch costs and job timelink constraints in a multi-objective context. Three of the most relevant objectives for cost reduction and production efficiency in a wafer fab are optimised, which notably translate into significant cost savings. As the benchmark study clearly showed, our proposed hybrid two-stage scheduling approach significantly outperforms existing traditional heuristic-based dispatch methods. Our scheduling solutions achieved on average a major reduction of more than 72% of violated timelink constraints, 39% on queuing time and an average reduction of 15% on the number of created batches. These improvements on the above schedule metrics are mainly a result of the iterative allocation of jobs to machines in reduced problem size and the better batching formations that efficiently exploit the timelink dependencies and batching costs. Our research and development activities will further focus on additional tuning of the lexicographic objective function, and more efficient decomposition methods to reduce the computational time needed to reach the same solution and provide a high solution quality for even larger sized problems.

ACKNOWLEDGMENTS

REFERENCES

Baez Senties, O., C. Azzaro-Pantel, L. Pibouleau, and S. Domenech. 2006. "Development of a multiobjective scheduler for semiconductor manufacturing". In *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, edited by W. Marquardt and C. Pantelides, Volume 21 of *Computer Aided Chemical Engineering*, 2015–2020. Elsevier.

Chan, F. T. S., T. C. Wong, and L. Y. Chan. 2006. "Flexible job-shop scheduling problem under resource constraints". *International Journal of Production Research* 44(11):2071–2089.

Defersha, F. M., and D. Rooyani. 2020. "An efficient two-stage genetic algorithm for a flexible job-shop scheduling problem

with sequence dependent attached/detached setup, machine release date and lag-time". *Computers Industrial Engineering* 147:106605.

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 Rose, O.; Uhrmacher, A.; Rabe, M.; Laroque, C.; Pasupathy, R.; and Himmelspach, J., editor(s), pages 1–10. Berlin, Germany: Institute of Electrical and Electronics Engineers, Inc.

Knopp, S., S. Dauzère-Pérès, and C. Yugma. 2017. "A batch-oblivious approach for Complex Job-Shop scheduling problems". *European Journal of Operational Research* 263(1):50–61.

Kopanos, G. M., D. Xenos, S. Andreev, T. O'Donnell, and S. Feely. 2020. "Advanced Production Scheduling in a Seagate Technology Wafer Fab". In *Proceedings of the 2020 Winter Simulation Conference*, edited by Bae, K.; Feng, B.; Kim, S.; Lazarova-Molnar, S.; Zheng, Z.; Roeder, T.; and Thiesing, R. editor(s), pages 1954–1965. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mönch, L., J. W. Fowler, and S. J. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities*. Springer New York.

Nattaf, M., S. Dauzère-Pérès, and W. C. Yugma C.. 2019. "Parallel machine scheduling with time constraints on machine qualifications". *Computers Operations Research* 107:61–76.

Sadeghi, R., S. Dauzère-Pérès, C. Yugma, and G. Lepelletier. 2015. "Production control in semiconductor manufacturing with time constraints". In *2015 26th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 29–33.

Senties, O. B., C. Azzaro-Pantel, L. Pibouleau, and S. Domenech. 2010. "Multiobjective scheduling for semiconductor manufacturing plants". *Computers Chemical Engineering* 34(4):555–566.

SIA 2021. "Global Semiconductor Sales Increase 6.5% to \$439 billion in 2020". https://www.semiconductors.org/global-semiconductor-sales-increase-6-5-to-439-billion-in-2020/, accessed 4th April 2021.

Sweney, M. 2021. "Global shortage in computer chips reaches crisis point". https://www.theguardian.com/business/2021/mar/21/global-shortage-in-computer-chips-reaches-crisis-point, accessed 4th April 2021.

Tamssaouet, M., S. Dauzère-Pérèsa, S. Knopp, A. Bitar, and C. Yugma. 2021. "Multiobjective Optimization for Complex Flexible Job-Shop Scheduling Problems". *European Journal of Operational Research*.

Yu, T., H. Kim, C. Jung, and T. Lee. 2013. "Two-stage lot scheduling with waiting time constraints and due dates". In *Proceedings of the 2013 Winter Simulation Conference*, edited by Hill, R.; Kuhl, M.; Pasupathy, R.; Kim, S.H.; and Tolk, A., editor(s), pages 3630–3641. Washington, D. C.: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

SEMYA ELAOUD is the R&D lead at Flexciton. She designs prototypes and experiment solutions for new product requirements. She is a PhD in Industrial Engineering and has 14 years of industrial experience dealing with a wide verity of optimization problems. She is the author of more than 25 scientific papers on multiobjective optimization, production scheduling, MILP and genetic algorithms. She is three times winner of the WBI excellence grant in the field of transportation and logistics. Her email address is semya.elaoud@flexciton.com.

RUARIDH WILLIAMSON is a Senior Optimisation Engineer in the commercial team at Flexciton with five years experience in optimization modelling, data science, and analytics consulting. He leads the technical solutions of early-stage proof of concepts for prospective clients, often creating new and novel solution strategies to meet their needs. He has completed a BSc and MSc in Operations Research (Distinction) at the University of Melbourne and London School of Economics respectively. His email address is ruaridh.williamson@flexciton.com.

BEGUN EFEOGLU SANLI is an Optimization Engineer at Flexciton. Her research interests include scheduling, supply chain management, facility layout planning, routing and logistics. She holds a Master of Science, Industrial Engineering degree from Middle East Technical University. She joined Flexciton 3 years ago after working in academia for a few years. Since then, she has been involved in developing solution strategies for Flexciton's planning and scheduling technology. Her email address is begun.efeoglu@flexciton.com.

DENNIS XENOS is the founding CTO at Flexciton. With over 10 year's experience in modelling, optimization and engineering, Dionysios is responsible for overseeing the optimization based technology being developed at Flexciton. He holds a PhD in Chemical Engineering from Imperial College London. He has innovated in the field of the operations optimization of large industrial plants and of the most complex manufacturing environments considering data-based models and predictive information combined with mathematical programming models. He is an author of more than 15 scientific publications and one patent of the optimization of industrial problems. His email address is dionysios.xenos@flexciton.com.