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

STUDY ON OPTIMIZATION POTENTIAL INFLUENCING FACTORS IN SIMULATION STUDIES FOCUSED ON PARALLEL BATCH MACHINE SCHEDULING USING VARIABLE NEIGHBOURHOOD SEARCH

Robert Kohn
Oliver Rose

Universität der Bundeswehr München
Institut für Technische Informatik
Fakultät für Informatik
Neubiberg, D-85577, Germany

ABSTRACT

Studies on operational lot scheduling in semiconductor manufacturing show significantly varying optimization potentials, depending on a multitude of factors relating to methods and models in simulation. We present experiments examining Variable Neighbourhood Search (VNS) used to improve the objectives queuing time and tardiness for the parallel batch machine scheduling problem. The discussed results incorporate the effects of specific model characteristics and constraints, namely incompatible job families, process dedication schemes, critical time bounds, and minimal batch size constraints among others. With regard to methodical factors, we examine the effect of time window decomposition on simulation results, and we discuss fundamental VNS settings, respectively their influence on improvements measured for problem instances of size relevant for industrial applications. This study intends to identify important factors in scheduling studies and evaluates their influence on optimization potentials based on extensive experiments.

1 INTRODUCTION

Semiconductor manufacturer's economic competitiveness relies to a not negligible extend on the shop floor control system. Effective material flow control policies enable companies to manage production accordingly to their goals, for example in terms of cycle times or customer delivery dates, constantly under pressure of cost reduction. Research and industry work intensely on scheduling topics in order to replace (state of the art) dispatching rule based systems by more powerful scheduling solutions, sooner or later. Scheduling solutions powered by optimization have been focused stronger than ever before in cause of effective search methods and constantly increasing computing power of modern computer systems. Operational scheduling promises to realize optimization potentials in production logistics unreachable for common dispatching systems. Especially (meta)heuristics seem to be practicable to tackle complex scheduling problems of dimensions interesting for practitioners in industry.

In this paper we examine the parallel batch machine problem in the furnace area. Here, a batch defines a quantity of jobs grouped together to be processed on a machine as one operation. The underlying model incorporates various constraints, to our knowledge all important for the industry, namely: process dedications, incompatible job families, minimal thresholds for batch sizes, and maximal time bounds.

The task is to optimize schedules for given objectives, respectively to minimize total queuing time (TQT) or total tardiness (TT). For that purpose, we apply the Variable Neighbourhood Search (VNS) approach, first mentioned by Hansen & Mladenović (1997). In view of NP-hard scheduling problems, Time Window Decomposition (TWD) is applied to disassemble scheduling problems, very similar to the rolling

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

horizon procedures presented in (Ovacik and Uzsoy 1995). We compare VNS with well known dispatching schemes; as references we consider First In First Out (FIFO) and Batched Apparent Tardiness Cost (BATC) discussed in (Balasubramanian et al. 2004).

Motivated by the observation in several studies, especially those based on real-world datasets, that optimization potentials in simulation vary strongly, we deem it necessary both to identify and evaluate the factors that have significant influence on the results. Especially our partner in industry needs reliable simulation results to asses operational scheduling for use on the shop floor; and knowledge about main factors and their influence on potentials increases confidence and trust in simulation results finally affecting decisions. We distinguish between influencing factors associated to the model. On one hand, a particular scheduling problem (instance), and on the other, those factors associated to the scheduling approach, relating to the search method or to the decomposition technique(s) applied.

Scheduling problems are commonly categorized, considering the material flow and the machines interrelationship respectively, while including the objective function and the constraints (Graham et al. 1979). A particular scheduling problem instance is additionally defined by a set of parameters and distributions, defining the solution space offering and limiting optimization potentials. Important characteristics are for example the number of machines and jobs, process dedication schemes and distributions of process times, batch sizes, job arrivals, and due dates. In the methodical area, we know that scheduling methods (especially their setup) in conjunction with decomposition techniques remarkably impact the results in simulation. Consequently we are particularly interested in important system variables that significantly influence the quality of simulation and optimization results.

Within the last years we implemented a simulation and optimization framework for operational lot scheduling, strongly oriented to the needs of semiconductor manufacturing. The framework covers an implementation of VNS used to improve schedules for given objectives, operating on an object oriented model structure representing the semiconductor manufacturing process. In addition to functionalities creating (and validating) scheduling problem instances from real world datasets, we implemented model generating procedures creating independent model instances for scheduling problems with specific characteristics. The time window shifting decomposition technique is applied in order to manage large scale problems, and to compete scheduling and dispatching strategies against each other.

For this paper, thousands of simulation runs were carried out, organized in six experiments. We convey a feeling for scheduling problem dimensions and computational complexity, pointing to the border of scheduling applications in real world. We give insights into the roles of particular VNS parameters, and explicitly point out their effect on optimization potentials. We give an overview about favourable and undesirable model characteristics, stressing leverages on benefits we expect to create with operational scheduling.

2 PROBLEM DESCRIPTION

We describe the considered problems by use of the $\alpha|\beta|\gamma$ classification scheme (Graham et al. 1979). Next, we introduce the notation to describe the considered scheduling problems.

- *Rm:* unrelated parallel machines (with unequal processing times),
- M_j*:* machine dedications (a job is dedicated to a restricted set of machines),
- r_j*:* nonzero release date of a job (dynamic job arrivals),
- *p-batch:* parallel batching (a number of jobs is processed simultaneously on a machine),
- *incompatible:* incompatible job families (jobs of different families cannot be processed together),
- b_j*:* arbitrarily (maximum) batch size for a job (family) on a machine,
- C_j*:* completion time of a job,
- d_j*:* operation due date of a job (equivalent to the initial tardiness),
- p_j*:* processing time of a job on a machine,
- *TT:* total tardiness defined as $\sum\max(C_j - d_j, 0)$,
- *TQT:* total queuing time defined as $\sum(C_j - r_j - p_j)$.

The studied (unrelated) parallel batch machine scheduling problems include constraints that experts in industry recognize as necessary to apply scheduling on the shop floor. We focus on the scheduling problem $Rm|M_j, r_j, p\text{-}batch, incompatible, b_j|TT$ that is NP-hard by reduction to the (NP-hard) problem $1||TT$ (cf. for example Lawler 1977). We also consider the problem $Rm|M_j, r_j, p\text{-}batch, incompatible, b_j|TQT$ that can be reduced to $1|r_j|\sum(C_j - r_j)$ for which Lenstra, Rinnoy Kan and Brucker (1977) showed NP-hardness; when considering that $TQT=\sum(C_j - r_j - p_j)$ is equivalent to $\sum(C_j - r_j)$ for constant p_j, then $Rm|r_j|TQT$ is equivalent to $P|r_j|\sum(C_j - r_j)$.

The underlying model optionally incorporates critical constraints, investigated in a separate experiment. First, we include time bounds for jobs, individually constituting maximal time spans for processing the jobs. Second, a minimum batch size defines a lower limit for the number of wafers in a batch, while lots count arbitrarily lot sizes up to 25 wafers.

3 RELATED WORK

The parallel batch machine problem is often studied for varying sets of constraints and objectives. Correspondingly to the scope of this work, we restrict ourselves to focus on literature that deal with (unrelated) parallel batch machine problems and approaches optimizing on-delivery measures (tardiness, lateness, etc.) and/or completion time (cycle time, flow time), incorporating at least incompatible job families and dynamic job arrivals.

Mönch et al. (2005) present a genetic algorithm (GA) combined with decomposition techniques using an extension of the Apparent Tardiness Cost (ATC) dispatching rule to minimize total weighted tardiness (TWT). Another GA approach for minimizing maximum lateness is proposed in (Malve and Uzsoy 2007). Li & Wu (2008) propose an approach minimizing TWT, based on the idea of Ant Colony Optimization (ACO). In (Klemmt, Weigert, Almeder & Mönch, 2009) a VNS approach is compared to a Mixed Integer Programming (MIP) solution combined with TWD (cf. Ovacik and Uzsoy, 1995), both minimizing TWT. Chiang et al. (2010) propose in their work a memetic algorithm to tackle the (unrelated) parallel batch machine problem with incompatible job families and job arrivals, while minimizing TWT.

We further refer to a scheduling approach based on the Next Arrival Control Heuristic (NACH) presented in (Ham and Fowler 2008); they describe a concept for scheduling wet etch and furnace operations using future job arrival information (look-ahead).

4 VARIABLE NEIGHBOURHOOD SEARCH

The concept of VNS, first described by Hansen & Mladenović (1997), thereafter adapted by several researchers for a multitude of applications, proposes the definition of problem specific neighbourhood structures disassembling large scale problems. A neighbourhood, representing problem specific knowledge, defines a specific kind of modification applied to a solution. Note that not every modification leads to a valid schedule due to the existence of critical constraints, e.g. time bounds. Each defined neighbourhood constitutes a smaller partial problem offering the possibility to find improved solutions in adequate time even for large combinatorial problems. Hansen and Mladenović (2001) describe two basic search schemes among others, Variable Neighbourhood Descent (VND) and Variable Neighbourhood Search (VNS).

VND repeats sequentially exploring neighbourhood structures (searching for the best neighbour) of an incumbent solution until no improvement is obtained. If a solution thus obtained (as a result of exploring a particular neighbourhood structure) outperforms the current solution, search restarts exploring neighbourhoods around the improved solution. Note that VND search is deterministic and always leads to identical results. The initial solution is obtained by FIFO or BATC, depending on the objective (see subsection 5.1) .

VNS combines a local search scheme improving the incumbent solution with the ability to escape from local optima by use of random movements in the solution space (shaking). Starting from an initial solution, the local search phase is continued until no improvement is obtained. Without knowledge about

the optimal solution (global optimum), we must assume that local search results always represent non-optimal solutions (local optima). In order to escape from a local optimum, the current solution is randomly modified in the shaking phase tolerating deteriorations, and subsequently followed by local search hopefully leading to a better solution. Our implementation of VNS applies VND for local search using identical neighbourhood structures for exploring and shaking. We defined four neighbourhoods combining two kinds of modifications (swap and move) operating on batches or jobs, similar to the neighbourhood structures presented in (Klemmt et al. 2009). The neighbourhoods are listed below in the sequence they were sequentially visited during local search and shaking phase. So far, we choose that sequence as result of our experiences, not as a result of reliable studies.

1. Swap Batch – Swaps the positions of two batches.
2. Swap Job – Swaps two jobs out of two batches.
3. Move Batch – Moves a batch to another place.
4. Move Job – Moves a Job to another batch.

Mönch et al. (2005) and Klemmt et. al (2009) both apply TWD (Ovacik and Uzsoy 1995), for their optimization approaches (GA and MIP) in order to create or improve solutions for subproblems. Both apply TWD in an event-based manner, meaning that every time a new decision is possible (for example a new job arrives, machine finishes processing, etc.) a new time window with fixed size is created, respectively a new subproblem. That time window (rolling horizon) frames future job arrivals that are taken into account when solving the subproblem; for that time span, we prefer to use the term *look-ahead horizon* throughout the rest of this paper. In contrast to that, the VNS approach in (Klemmt et al. 2009) uses full information about job releases without TWD, and therefore has advantage over MIP (using TWD) by an unlimited look-ahead horizon. Our implementation of TWD works not necessarily event-based, instead, we are able to define the time window shifting interval arbitrarily. As a consequence, we are able to apply VND/VNS with or without TWD, or with arbitrarily time window intervals.

5 DESIGN OF EXPERIMENTS

We defined six experiments, where each experiment varies up to three parameter in order to evaluate their effect on TQT or TT. The experiments were carried out on a DELL Blade Server with eight computing nodes, each with two Quad-Core E5450 Xeon CPUs (2,8Ghz) and 16GB memory per node, operated by Microsoft HPC Windows 2008 (64bit); in total we had 64 cores running in parallel most of the time.

5.1 Overview

Tens of thousands simulation runs, organized in six experiments, generated the results discussed in this paper. Decomposition techniques were often used to manage complexity of focused scheduling problems, e.g. time window shifting (rolling horizon). Here, were are particularly interested in the length of time windows and their effect on the results. Several studies showed that additional information about job arrivals influence optimization results, correspondingly we examine different widths for the look ahead horizon. One experiment deals with model complexity respectively scheduling problem size, here defined by the number of machines, the number jobs and the level of utilization. A model instance is additionally characterized by its process dedication scheme, the number of job families, and the distribution of process times; we defined an experiment to examine the impact of these factors. Since we conduct research in close collaboration to industry, we face critical constraints as time bounds and minimal batch sizes, and so we investigate the effect of both constraints combined with varying distributions for lot sizes. In contrast to already mentioned experiments mainly examining model characteristics, we defined an experiment focused on VNS that investigates more detailed the shaking effect in conjunction with varying deadlines for computation time for three exemplary model sizes.

Table 1: Design of Experiments

Experiment	Objective	Dispatching reference	Time Window Decomposition	Investigated parameters
TWD	TQT	FIFO	Used	Time window shifting interval
Look Ahead	TQT	FIFO	Used	Look ahead horizon
Model Complexity	TT	BATC	Used	Number of machines
				Number of jobs
				Utilization level
Model Characteristics	TT	BATC	Used	Dedication scheme
				Number of job families
				Distribution of process times
Critical Constraints	TQT	FIFO	Used	Threshold for time bounds
				Minimal batch size
				Distribution of lot sizes
VNS	TT	BATC	Not Used	VNS method (shaking on/off)
				Computing time deadline

5.2 Model Default Settings

A model generator creates model instances with specific characteristics. We defined a set of important characteristics that describe important viewpoints of a scheduling problem instance.

The default model instance states the problem to schedule 1500 jobs on 30 machines at utilization level 0.8. We define 30 incompatible job families, that is realistically as many as available machines, triangularly dedicated to the machine pool (see section 6.4). The process times are uniformly distributed between four and eight hours. The batch size varies between four and eight lots, depending on job family and machine. We include no critical constraints e.g. time bounds and minimal batch sizes for the default model, only within the experiment "Critical Constraints". The default model contains only jobs respectively lots with maximal lot size (25 wafers). Job arrivals are distributed uniformly from zero to the expected makespan. The due dates are normally distributed relatively to the arrival date with standard deviation of twelve hours.

Table 2: Basic default model settings

Machines	Jobs	Utilization
30	1500	0.8

Table 3: Facility specific default model settings

Dedication	Job families	Process times	Max batch size	Min batch size	Time bounds
Triangular	30	U~(240,480) [min]	U~(4,8) [lot]	None	None

Table 4: Job specific default model settings

Lot size	Job arrivals	Due dates
Const~(25) [wafer]	U~(0,makespan)	N~(0,12) [hours]

5.3 Method Default Settings

By default we apply identical settings to parameterize the optimization method. Experiments only deviate from defaults in those cases in which a certain parameter (or set of parameters) is within the scope of the study's investigation and therefore varied. With exception to the experiment "VNS", we apply the TWD technique with an interval length of ten minutes. As default for the look ahead horizon we use 30 minutes,

which means that for every time window the information about job arrivals within next 30 minutes is available for optimization. By default we apply the VNS derivate Variable Neighbourhood Descent (VND) that only executes the local search phase without shaking, stopping when the first optimum is reached, but limited to 60 seconds as computing deadline for each time window. However, VND never took more than five seconds per time window on average measured for the model default settings. We use dispatching policies in simulation to generate initial solutions, subsequently improved by VND. For those experiments with an objective function minimizing TQT we use FIFO dispatching as initial solution; remaining experiments focused on TT start their search from solutions generated with BATC.

Table 5: Default Method Settings

Decomposition Technique	Time window shifting using ten minutes as interval
Look Ahead Horizon	Constantly 30 minutes (1/12 average process time)
Search Method Type	Variable Neighbourhood Descent (VND)
Computational Deadline	At maximum 60 seconds per time window (never reached)
Initial Solution	FIFO for TQT as objective function
	BATC for TT as objective function

5.4 Exception for Default Settings

The experiment "VNS" is an exception with regard to the default settings for the model as well as for the method. In this case time window shifting was not applied and therefore significantly smaller models need to be focused, where each of three focused models constitutes a scheduling problem for which VNS and VND still lead to improved schedules in adequate time.

5.5 Reliability and Replications

For every model description, having a specific characteristic, we generated 20 independent instances. Since VNS (using shaking) relies on randomness, consequently shows stochastic effects, we repeat those runs ten times to reach a certain statistical reliability. Of course deterministic simulations runs (VND or dispatching) were only done once, since these always lead to identical results due to their deterministic behaviour.

6 RESULTS OF EXPERIMENTS

In this section we discuss the results of the experiments. We give VNS/VND improvements in relation to the dispatching results generated by FIFO or BATC. For the entire analysis and for all experiments, we discuss the average value for queuing time or tardiness calculated from the schedules generated in each simulation run.

6.1 Time Window Decomposition

Time window shifting decomposition is an essential technique used to disassemble scheduling problems on the timeline into smaller sub-problems, each sequentially separately solved by optimization. This way it is possible to evaluate scheduling methods for arbitrarily large problem instances with practical relevance. This experiment examines the effect of time window decomposition, more precisely the width of a window, on TQT. This experiment shows that the results respectively the optimization potentials remarkably rely on time window size, varied from 1 to 240 minutes in this experiment (see Figure 1).

One might assume that the smallest time window, considered synonymous to an event based simulation, would lead to best results. The experiment confirms this statement in so far as the queuing time tends to increase with increasing time window size. Obviously, the greater the time window interval value, the less the probability to start a certain job or batch just in the moment of its arrival. In other words, jobs already arrived were not scheduled to start before the next time window begins.

But, we also observed the effect (for most model instances) that there exists at least one value for the time window size leading to considerably better results than the smallest possible time unit does. Since we deal with discrete events, discrete job arrivals over time, never identical for different model instances, it is clear that varying time window sizes lead to varying results. We state that the time window size in conjunction with randomly distributed job arrivals strongly effects the results, and the smallest time window does not necessarily lead to the best result with respect to the focused objective.

It is also remarkable that applying an optimization technique does not lead to global improvements for all the time window interval values tested in this experiment. We observe the phenomenon that there are values for the time window interval that lead to "optimized" solutions outperformed by their corresponding dispatching reference, which is obtained by the identical dispatching rule used to create the initial solution. TWD disassembles the given scheduling problem and thus creates a sequence of subproblems solved and optimized sequentially. Each scheduling decision taken within a time window (subproblem) has an effect on the next subproblem in sequence, and remains effective for all succeeding subproblems. The key point is that an optimized solution for a single subproblem may cause unfavourable situations (compared to the non-optimized dispatching solution) leading to a sequence of subproblems where optimization does not compensates the early scheduling decisions. "It is hard to take the best decision if you do not have all the information." This effect is triggered by the interrelationship between job arrivals and interval length, pointing to the discrete nature of the entire system as well as to a structural weakness of scheduling approaches using TWD. The improvements realized by applying VND vary between plus/minus ten percent, although the improvement on average is positive.

We state that studies, meant to generate reliable statements on optimization potentials, need to frame a sufficiently high number of independent model instances evaluated with varying time window sizes when TWD is applied. Especially for industrial applications (or cost-benefit calculations via simulation) it is hard to gather an adequate number of real-world models from history in order to evaluate scheduling methods discussed to be introduced to production systems.

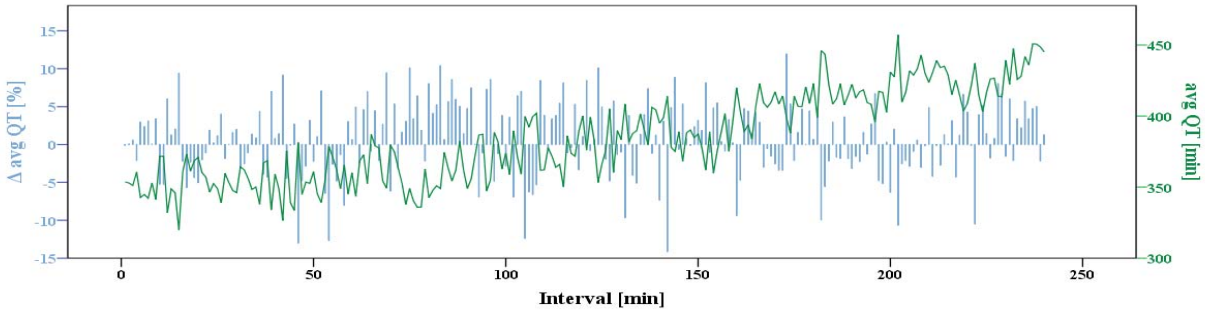

Figure 1: Average queuing time (improvements) for an exemplary model instance

6.2 Look Ahead

The term "Look Ahead" in the scope of scheduling stands for additional information about job arrivals in the future seen from the current point of time at which a scheduling problem occurs, in simulation systems as well as in industrial applications. The look ahead horizon defines the time span to which arising job arrivals were taken into account when optimizing the schedule for a given objective. This study shows that the look ahead horizon strongly effects scheduling results, and especially the improvements gained by optimization.

The results show that information about soon arriving jobs empowers the scheduling method remarkably, most likely due to the fact that the examined scheduling problem incorporates the batching functionality. Figure 2 shows a clear trend to increasing improvements accompanied with an increasing look ahead horizon. Incorporating look ahead information to the scheduling system offers the opportunity to delay waiting batches, which could already be scheduled to start, in order to wait for jobs arriving soon.

Furthermore, the optimization procedure will probably schedule small batches to start immediately having the knowledge about no near future job arrivals.

The results show that the improvement gained by VND compared to FIFO dispatching tends to steadily increase with increasing look ahead horizon, up to 40% in the area of 480 minutes used for the look ahead horizon. Remember that the process time is uniformly distributed between 240 and 480 minutes.

It is remarkable that the improvement shows also negative improvements down to minus ten percent for individual look ahead horizon values smaller than 60 minutes (observed for this particular model instance). The queuing time improvement shows fluctuations for neighboured horizon values, again because of the discrete structure of model instance and simulation method.

In addition to that we observed in further studies examining greater values for the look ahead horizon (not presented here) that the improvement reaches a plateau for values greater than one average process time for the model, and holds that improvement level up to two times average process time before the improvement decreases again in cause of computational complexity of arising scheduling problems.

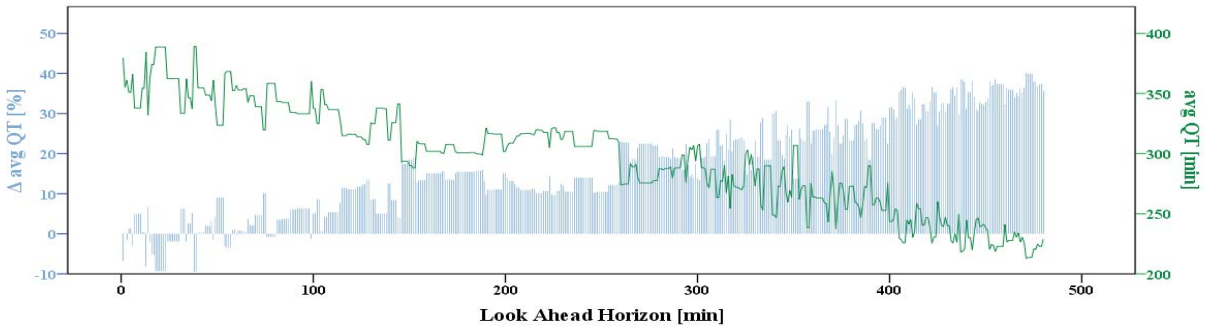

Figure 2: Average queuing time (improvements) for an exemplary model instance

6.3 Model Complexity

A scheduling problem's complexity is significantly defined by the number of jobs and the number of machines. If TWD is used, the utilization level determines the number of jobs creating the scheduling problem in conjunction with available machines. The higher the utilization level, the higher the density of job arrivals per timeframe. This experiment evaluates 27 different model descriptions, each characterized by differing value combinations for the number of machines, the number of jobs and for the level of utilization. The objective function is defined for minimizing TT of the jobs.

Table 6: Varying Parameter for the Experiment "Model Complexity"

Parameter	Value Range
Machines	{ 15, 30, 60 }
Jobs	{ 300, 600, 1200 }
Utilization	{ 0.7, 0.8, 0.9 }

Figure 3 shows the VND improvements for the average tardiness, visualized with blue bars. The improvements in average tardiness vary between 1 and 17% compared to the results generated with BATC dispatching. The diagrams suggest that the average tardiness improvements seem to increase with increasing number of machines and decreasing utilization level. It is likely that a higher number of machines results in increased alternatives offering space for improvements. We also consider it not implausible that delaying batches for incoming jobs takes more effect at lower utilization levels. We also observed that the absolute average tardiness values, visualized with green filled circles in the diagram, increase with increasing utilization level, which was expected. We like to note that the average runtimes per time window never reached the upper limit set to 60 seconds for none of the model instances.

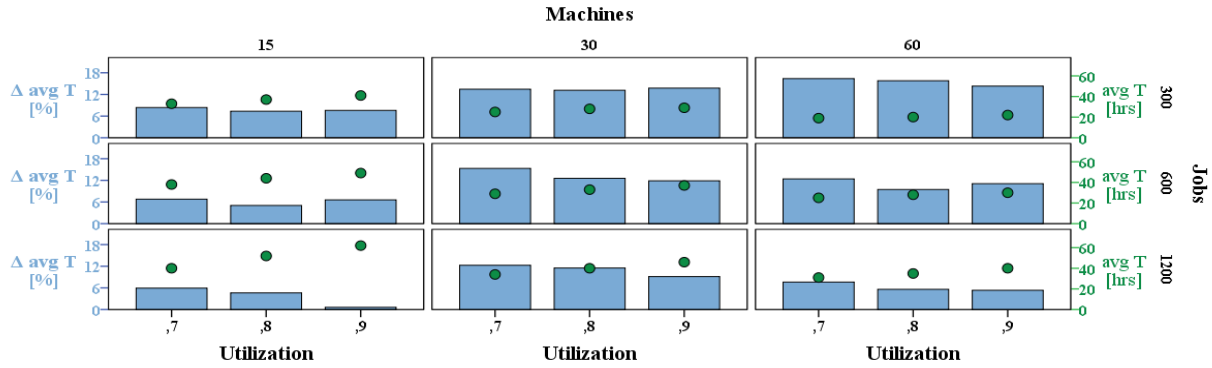

Figure 3: Average tardiness (improvements) depending on model complexity

6.4 Model Characteristics

This experiment provides insights in the influence of different process dedication schemes, about the impact of job families and about the effect of varying process times respectively their distribution. The process dedication scheme represents an exemplary strategy to qualify machines for processes. "OOAK" (one of a kind) stands for a dedication scheme in which machines only provide a single process. Conversely, there exists only one machine for each process respectively job. "Uniform" represents the unrestricted scheme in which all processes were dedicated to all machines without limitations. "Triangular" schemes define one single machine of type OOAK and one single machine of type Uniform. Imagine a lower/upper triangular matrix where each non-zero matrix cell represents a specific process dedicated to a certain machine. We e set the job families to 15, 30 or 60.

Restricting model characteristics in this experiment, process dedications and job families, also limit optimization potentials. OOAK dedication schemes show comparatively lowest improvements tending to zero, whereas uniform dedication schemes show the best performance values. Similar to the effect observed for restricted dedication schemes we monitored improvements tending to decrease with increasing number of job families, while absolute tardiness values increase simultaneously. The results also suggests that there exists a relationship between improvements and process time distributions; the improvements seem to increase with increasing spread of process times. Figure 4 shows the average tardiness improvements with blue bars and absolute values for average tardiness using green circles.

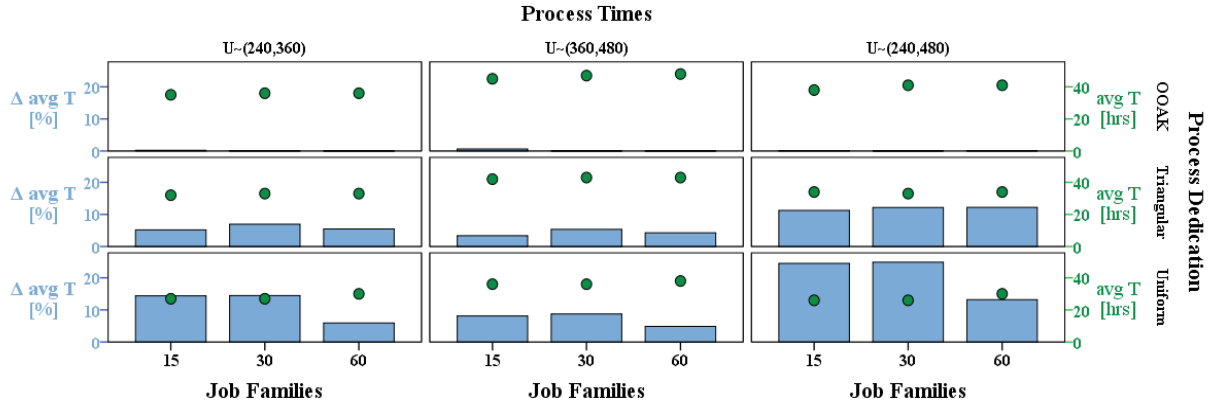

Figure 4: Average tardiness (improvements) depending on model characteristics

6.5 Critical Constraints

We also examine the influence of critical constraints on optimization potentials, that are time bounds and minimal batch size thresholds in this experiment. A job/lot is oxidizing while waiting under atmospheric conditions for the next process in the production cycle. A time bound defines a timeframe for a job to process the next step in order to prevent from violations of process quality specifications caused by oxidation, which are more likely to occur when the limited waiting time is exceeded. In case of violated time bounds, experts have to individually evaluate the situation for the affected job, disturbing the continuous material flow. The minimal batch size constraint defines a lower threshold in wafers for a batch of jobs, depending on the process job family and the machine.

Both constraints may lead to critical situations, in simulation as well as in reality, in which jobs are unscheduled; either there exists no valid solution or the method does not find a valid schedule including critical jobs. This means with regard to optimizing heuristics that changes made to valid solutions during search do not always lead to feasible solutions. The focused experiment varies maximal limits for time bounds and minimal limits for batch sizes, accompanied with varying distributions for the lot size.

Table 7: Varying Parameter for the Experiment "Critical Constraints"

Parameter	Value Range
Time Bounds	{ U~(240,360), U~(360,480), None }
Min Batch Size	{ None, U~(0,0.25), U~(0.25,0.5) }
Lot Size	{ U~(1,25), U~(15,25), Const~(25) }

Figure 5 shows percentages of unscheduled jobs with blue bars as well as the average queuing time using green circles. We see that tight time bounds, here represented by uniformly distributed bounds between 360 and 720 minutes, lead to ten percent non scheduled jobs in the experiment. But the experiment also shows positive effects of tighter time bounds on the average queuing time. For the minimum batch size constraint, uniformly distributed between 0.25 and 0.5 times the maximum batch size, we observed up to five percent unscheduled jobs because of missing suitable batch partners in the experiment.

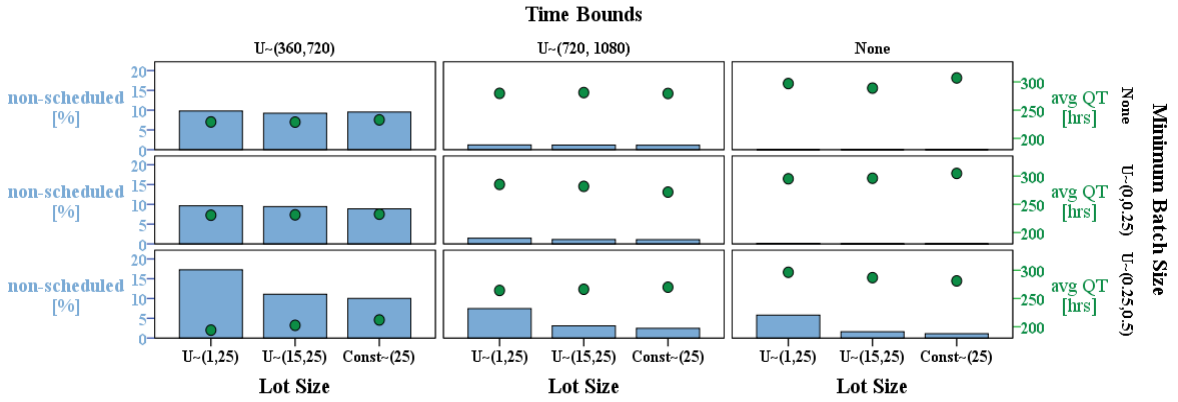

Figure 5: Non-scheduled jobs and average queuing time under consideration of critical constraints

In cooperation with our industry partners, we made the experience that objectives and constraints are sometimes mixed in minds and systems, so the minimal batch size threshold is commonly considered as constraint. On the one hand the furnace process requires a minimum number of wafers inside the process chamber (boat) in order to reach specified process parameters. On the other hand there always exists the possibility to fill up a batch with non-productive wafers until a batch condition is satisfied. The urgent intention to define minimal batch size thresholds, especially in dispatching systems, is to prevent from running too small batches that would result in a loss of throughput, and in consequence to reduced cycle

times. In the scope of optimization it is highly beneficial to define minimal batch size thresholds as low as possible, increasing the degree of freedom for scheduling decisions. Concerns regarding performance indicators in production logistics are then only a question of properly defined objective functions.

6.6 VNS

On the basis of this particular experiment we compare VND and VNS for three model sizes. VND stops the search reaching a local optimum, within this scope defined by a solution where no single neighbourhood movement leads to an improvement. VNS performs random changes (shaking) tolerating deteriorations in order to restart VND as local search, continuing shaking and local search alternately until the temporal deadline is met. We are particularly interested in the effect of the random changes in the VNS scheme, enabling local optimum escapes, and hereby we try to evaluate the gap in optimization potentials existing between VNS and VND. We tested three problem sizes with 5, 10 and 15 machines and each 20 times the number of jobs (100, 200, 300) to be scheduled without time window shifting decomposition under varying temporal deadlines up to 30 minutes.

Table 8: Varying Parameter for the Experiment "VNS"

Parameter	Value Range
Method	{ VNS, VND }
Deadline	{ 1, 2, 3, …, 30 } [min]
Machines/Jobs	{ [5,100], [10,200], [15,300] }

Figure 6 shows with box plots representing the distribution of tardiness VND/VNS improvements (compared to BATC) gained for 20 independent model instances under varying computational deadlines, evaluated for 3 models differing in their size. We state that the deviations in improvements decrease with increasing model size. For the smallest model with five machines we observe varying improvements up to 25%. The results also show that VNS outperforms VND whenever the initial local search phase was not aborted by reaching the computational deadline. For the smallest model (five machines and 100 jobs) VND reaches its local optimum within 60 seconds and VNS nearly performs the same for all covered deadlines. The midsize model shows that VNS and VND perform equally to the deadline of approximately five minutes, the time both algorithms need to finish the initial local search phase leading to the first local optimum; thereafter VNS passes to the shaking phase and leads to slightly better results on average. For the large scale model framing 15 machines, we observe no difference between VND and VNS, both variants do not overcome the local search phase within 30 minutes.

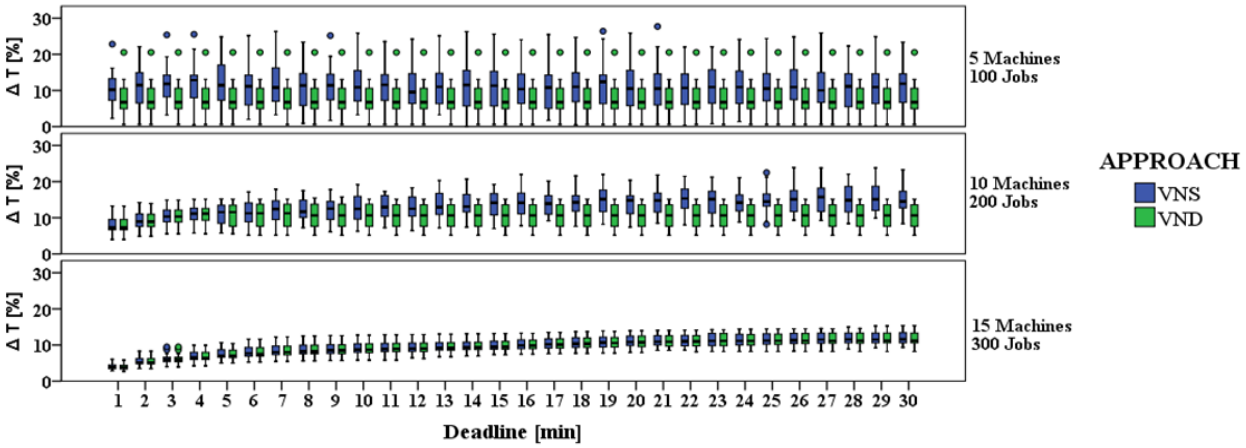

Figure 6: Average tardiness improvements compared with VND and VNS

7 CONCLUSIONS

Applying decomposition techniques is linked with uncertainties with regard to the optimization potential. Time window shifting, especially the interval width, has partly strong effects on the results. From the experiments it can be seen that decreasing intervals tends to lead to better results on average, but not necessarily for a single model instance. Small changes in decomposition intervals may lead to large changes in results.

Emphasizing the discrete character of focused scheduling problems we point to the fact that two independent model instances, both following identical model descriptions used for their generation, may show very differing results. In order to derive trustable statements from simulation in the area of scheduling it is necessary to consider a sufficient number of model instances and replications

Incorporating future job arrivals into the decision making process (scheduling) empowers optimization remarkably, at least for the focused parallel batch machine problem. Especially for the application in the furnace area with long process times of several hours, the decision when to start a batch is important in two respects. First, it is often beneficial to delay a batch in order to wait for one or more lots arriving soon. Second, it is better to start a smaller batch immediately, having the knowledge that no more lots arrive soon. For optimization it is beneficial to keep the scope for decision-making as large as possible, meaning that any kind of restriction, non-uniform process dedications or incompatible job families, limits the optimization potential.

With regard to computational complexity, based on our experimental results, we state that real-world applications, that may frame 60 parallel batch machines, constitute solvable scheduling problems for which VND leads to significant improvements within few seconds on average. VNS significantly performs better than VND, showing the effectiveness of implemented functionalities that enable the search to successfully escape from local optima.

Time bounds as a critical constraint cause invalid solutions along the heuristic optimization process. In order to reduce the number of unscheduled jobs we propose to apply time bound oriented dispatching rule prioritizing jobs near to the given time limit, instead of FIFO as we did in the experiment. Minimal batch size thresholds, considered as constraints rather than objectives, should be defined as low as possible, also increasing the degree of freedom for scheduling decisions.

LITERATURE

Chiang, T.-C., Cheng, H.-C., & Fu, L.-C. 2010. A memetic algorithm for minimizing total weighted tardiness on parallel batch machines with incompatible job families and dynamic job arrival. Computers & Operations Research, 37, 2257–2269.

Klemmt, A., Weigert, G., Almeder, C., & Mönch, L. 2009. A comparison of MIP-based decomposition techniques and VNS approaches for batch scheduling problems. In M. Edited by D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, & R. G. Ingalls, In *Proceedings of the Winter Simulation Conference 2009* (pp. 1686–1694).

Ham, M., & Fowler, J. W. 2008. Scheduling of Wet Etch and Furnace Operations with Next Arrival Control Heuristic. The International Journal of Advanced Manufacturing Technology, 38, 1006–1017.

Li, L., Qiao, F., & Wu, Q. 2008. ACO-Based Scheduling of Parallel Batch Processing Machines with Incompatible Job Families to Minimize Total Weighted Tardiness. In M. Edited by Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle, & A. Winfield. Ant Colony Optimization and Swarm Intelligence (pp. 219–226): Springer Berlin / Heidelberg.

Malve, S., & Uzsoy, R. 2007. A genetic algorithm for minimizing maximum lateness on parallel identical batch processing machines with dynamic job arrivals and incompatible job families. Computers & Operations Research, 34, 3016–3028.

Mönch, L., Balasubramanian, H., Fowler, J. W., & Pfund, M. E. 2005. Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. Computers & Operations Research, 32, 2731–2750.

Hansen, P., & Mladenović, N. 2001. Variable neighborhood search: Principles and applications. European Journal of Operational Research, 130, 449–467.

Mladenović, N., & Hansen, P. 1997. Variable neighborhood search. Computers & Operations Research, 24, 1097□1100.

Ovacik, I. M., & Uzsoy, R. 1995. Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. International Journal of Production Research, 33, 3173–3192.

Graham, R., Lawler, E., Lenstra, J., & Rinnooy Kan, A. 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. In P. Edited by Hammer, E. Johnson, & B. Korte. Discrete Optimization II Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver (pp. 287–326): Elsevier Science.

Lawler, E. L. 1977. A "Pseudopolynomial" Algorithm for Sequencing Jobs to Minimize Total Tardiness. In P. Edited by Hammer, E. Johnson, B. Korte, & G. Nemhauser. Studies in Integer Programming (pp. 331–342): Elsevier Science.

Lenstra, J. K., Rinnooy Kan, A. H. G., & Brucker, P. 1977. Complexity of Machine Scheduling Problems. In P. Edited by Hammer, E. Johnson, B. Korte, & G. L. Nemhauser. Studies in Integer Programming (pp. 343–362): Elsevier.

ACKNOWLEDGEMENTS

We like to thank our colleges from Infineon Technologies for their support at any time. Special thanks go to Jens Doleschal and Manfred Benesch from Dresden University of Technology who maintain server and systems essential for our experiments with all their expertise and commitment. This work was supported by Grant 13N11588 of the Federal Ministry of Education and Research Germany.

OLIVER ROSE holds the Chair for Modelling and Simulation at the University of the German Federal Armed Forces Munich, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from Würzburg University, Germany. His research focuses on the operational modelling, analysis and material flow control of complex manufacturing facilities, in particular, semiconductor factories. He is a member of IEEE, INFORMS Simulation Society, ASIM, and GI. His e-mail is oliver.rose@unibw.de.

ROBERT KOHN is a member of the scientific staff of Prof. Dr. Oliver Rose at the Chair of Modelling and Simulation at the University of the German Federal Armed Forces Munich, Germany. His focus is on simulation based scheduling in semiconductor manufacturing. He received his M.S. degree in computer science from University of Applied Sciences Stralsund, Germany. His e-mail address is robert.kohn@unibw.de