Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo and M. Loper, eds.

SIMULATION MODEL SIMPLIFICATION FOR CHANGING PRODUCT MIX SCENARIO

Igor Stogniy

Institute of Applied Computer Science
Technische Universität Dresden
Dresden, 01062, GERMANY

Wolfgang Scholl

Infineon Technologies Dresden
Königsbrücker Straße 180
Dresden, 01099, GERMANY

Hans Ehm

Infineon Technologies AG
Am Campeon 1-15
Neubiberg, 85579, GERMANY

ABSTRACT

Infineon Technologies Dresden has long used simplified simulation models to optimize production planning. However, the simplification is based on the gut feeling of experts who do not have time to analyze the various concepts in detail. In this paper, a detailed analysis of the simulation model simplification by substituting operations for constant delays was performed under conditions close to the real world. A statistical model was developed to calculate the delay values. The simplification results based on the statistical model are compared with the results based on the detailed model. The experiments were carried out based on the MIMAC dataset 5 model.

1 INTRODUCTION

This research was inspired by one of the Winter Simulation Conference (WSC) 2019 reviewers who wrote: "my understanding is that the reduction approaches only work for a given workload..." and probably the same reviewer at WSC 2020 who wrote, "I am still not fully convinced that reduction approaches can work well." Therefore, this paper considers the problem of building simplified simulation models in conditions close to real ones, i.e., when using a changing product mix. For this purpose, it is proposed to use a simplified (hybrid) model, in which a part of the tool sets is simulated in detail, and the other part is substituted for delays derived from a statistical model (metamodel). Usually, metamodels are used to replace the entire simulation model (e.g., Albey et al. 2017). The main feature of this paper is the joint use of detailed simulation model elements and metamodel elements. If further developed, we believe that this approach will help to perform optimization based on such simplified models. This paper's main contribution is that using the proposed statistical models does not lead to a significant deterioration in the accuracy of the simplified models compared to the simplified models based on the detailed model.

Indeed, in previous papers, we considered a steady-state (static product mix), which is relatively rarely used in the industry. However, it was suitable for working out a method to simplify simulation models by substituting tool sets for constant delays and for analyzing the accuracy measurements used. This paper discusses the dynamic product mix and has the following goals: 1) to describe the simulation model simplification approach, which uses a statistical model that predicts the behavior of a detailed model in the future (scenario α_2); 2) compare the results of the developed approach (α_2) with the baseline (scenario α_1 – using a detailed model to get information about its future behavior). Note that the primary goal of the considered simulation model is to predict lot cycle times, but not the behavior of the tool sets. As in previous

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

papers, MIMAC dataset 5 (MIMAC datasets 1997) was used. Due to many experiments to date, it has only been possible to carry out investigations for the FIFO dispatching rule. Autosched AP from Applied Materials, version 11.5, was used for modeling. R (R Core Team 2019) was used for output data analysis.

This paper is organized as follows. Related works are presented in Section 2. The design of the experiments, including the creation of initiating data and description of the α_2 scenario, are given in Section 3. The specific features of the α_1 and α_2 comparison are described in Section 4. Section 5 includes experimental results.

2 RELATED WORK

A review of papers on simulation model simplification can be found in (Van der Zee 2019) and (Mönch et al. 2018). Let us mention the papers that served as sources of inspiration and ideas for our experiments. Hung and Leachman (1999) reasoned for using constant delays rather than random delays for simplification because of the negative correlation in cycle time between subsequent process steps. Therefore, we use constant delays in our experiments. Völker and Gmilkowsky (2003) introduced the concept of a sieve function, which we also use in our work, increasing the number to ten. The sieve function is a work center criticality index according to which the given tool sets are ordered for substitution for delays. Sprenger and Rose (2010) conducted a series of experiments to obtain information about the behavior of the detailed model under different workloads, which they used to build a simplified model. Therefore, we considered the different workloads for different tool sets when building a forecasting model. In one of the figures, Jain et al. (2000) showed a dependence of cycle time on model time, which inspired using aggregated indices to evaluate the models' accuracy. Duarte et al. (2007) presented a parametrization methodology for supply chain nodes, which was then extended by Ewen et al. (2017) for the entire supply chain. Its essence is the use of empirical distribution of the capacity for simplification. The statistical model we developed is a further evolution of this thought. One of the differences in our research is using MIMAC dataset 5 with 11 products instead of MIMAC dataset 1 (2 products). This was probably the reason for the need to develop a more complex statistical model in our case.

There are many ways to build a forecasting model (Wang et al. 2020). It is interesting to note that Wang et al. (2020) consider big data analytics as an alternative to simulation. There is also a view where simulation is the source of data for training a neural network, which is then used for prediction (Huang et al. 2016). This paper proposes using a simulation model to build a forecasting model, which is then used again to build a simplified model. A similar approach can be found in the literature for constructing clearing functions (linear programming). An extensive review can be found in (Missbauer and Uzsoy 2020. For example, Albey et al. (2017) considered the use of multi-dimensional clearing functions. However, they used a very simple model consisting of 6 machines, 4 products, 12 operations without batching. On the other hand, we consider MIMAC5, which has 83 tool sets (174 machines), 11 products, 2178 operations, and batching. Therefore, it can be assumed that the method we developed is easier to adapt to large simulation models. However, the main reason for developing our own method was the need to integrate it with previous developments in simulation model simplification. This was easier to do when using statistical models. Besides, Infineon already has experience in using such models in real production planning (Beeg 2004), and it will be easier for us to convince the management of the solution feasibility. To evaluate the accuracy of the simplified models, we used the results of (Stogniy and Scholl 2020).

3 DESIGN OF EXPERIMENTS

To implement the experiments in this work, the previously developed automated experimental environment was used. This allowed to build graphs based on 6640 experiments to analyze simulation model simplification, such as those given at the end of this paper (five seeds for each experiment, the total sum of the simulation model runs is 33200). Each experiment represents the substitution of one tool set for a constant delay. Additionally, 14000 runs were performed to calculate delays (2000 for scenario α_1 and 12000 for scenario α_2). Because of the large volume of experiments, it was possible to carry out experiments

only with the FIFO dispatching rule. For a detailed description of the design of the experiments, see (Stogniy and Scholl 2020). This section describes mainly the important additions.

In this study, we used the following sieve functions: ζ_1 = *IDLE%*; ζ_2 = *IDLE% + PROC% – PROC%*(BS_{AVG} / BS_{MAX}); ζ_3 = *(100 – IDLE%) / IDLE#*; $\zeta_4 = QT_{AVG}$; $\zeta_5 = QT_{AVG} / PT_{AVG}$; $\zeta_6 = QL_{AVG}$; $\zeta_7 = QL_{AVG} / BS_{MAX}$; $\zeta_8 = CT_{SD}^{total}$; $\zeta_9 = CT_{SD}^{total} / CT_{AVG}^{total}$; and $\zeta_{10} = TH$. The following model statistics based on weekly standard model reports were used: *IDLE%/IDLE#* – the percent of time/the number of times a tool set entered the idle state; *PROC%* – the percent of time a tool set entered the processing state; BS_{AVG} – the average of batches processed (batch size); BS_{MAX} – the maximum quantity of pieces allowed in a batch; QT_{AVG} – the average time lots waited at the tool set (queue time); QL_{AVG} – the average number of pieces in front of the tool set (queue length); PT_{AVG} – the average of the lot processing time for the tool set; CT_{AVG} – the average lot cycle time for the tool set ($CT_{AVG} = PT_{AVG} + QT_{AVG}$); CT_{SD} – standard deviation of the cycle time for the tool set ($CT^2_{SD} = PT^2_{SD} + QT^2_{SD}$); TH – throughput for the tool set (Stogniy and Scholl 2020).

As in the previous paper, process step based delays (η_1) were also used. But instead of tool set based delays (η_2), its improved version, hybrid delays (η_3), was used:

$$\eta_3^X = \frac{RPT_X}{w.avg.RPT}\eta_2 = \frac{RPT_X \cdot \sum_i^n lot\_num_i}{\sum_i^n (RPT_i \cdot lot\_num_i)}\eta_2$$

where RPT_X is Raw Processing Time of X^{th} process step which belongs to the given tool set which has tool set based delay η_2; lot_num_i is the number of lots/week for ith process step; n is the total number of process steps corresponding to the given tool set; *w.avg.RPT* is weighted average RPT (used in section 3.2.2). Since we consider a dynamic product mix, the delays were calculated for each week of model time.

3.1 Making input data

The MIMAC model contains only data for the steady-state. Therefore, it is necessary to develop the input data by ourselves. For this purpose, a specially developed algorithm was used, which consists of the following. In the first 10 weeks, there are no changes (warm-up period). In week 11, two process flows are selected (14 and 15), for which the number of lots per week will decrease, and process flow 8, for which the lots/week will increase. In subsequent weeks the next process flows are chosen to increase (week numbers were selected manually). Each week $IDLE_{OEE}$ is recalculated for each tool set so that the minimum value is in the range (0.5%, 6.0%). $IDLE_{OEE}$ is the sum of operational and rate efficiency losses in terms of (SEMI E79-0200, 2000). This ensures high model load on the one hand and prevents overloading on the other hand. Then the algorithm moves the process flow from the list of increasing to the list of decreasing if it exceeds 31 lots/week or if it cannot grow further for a long enough time (several algorithm work cycles). The results are presented in Figure 1.

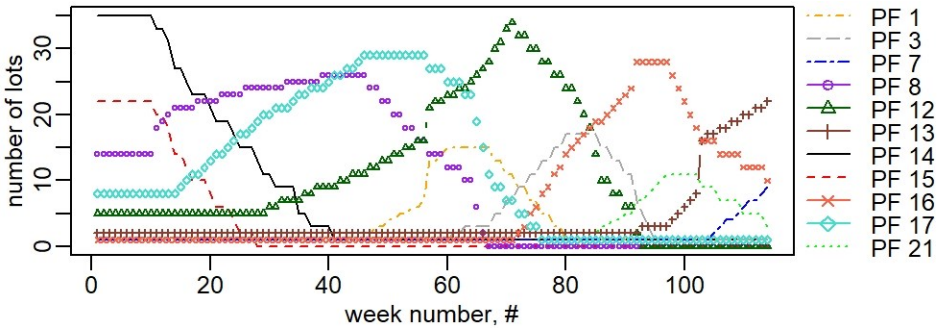

Figure 1: Experimental input data.

In the MIMAC5 model, 79 of 83 tool sets have no setup. Therefore, the following is applied to them:

$$IDLE_{OEE} = 1 - \frac{MTTR}{MTTR + MTTF} - PROC_{OEE} = 1 - \frac{MTTR}{MTTR + MTTF} - \sum\nolimits_i^n \frac{lot\_num_i \cdot (RPT_i + L)}{BSmax \cdot TS\_num}$$

where for a particular tool set *MTTR* – Mean Time To Repair, *MTTF* – Mean Time To Failure, *L* – Load time, *BSmax* – maximum batch size, *TS_num* – number of the machines in the tool set; lot_num_i – number of lots/week for i^{th} process step; *n* – total number of process steps for the given tool set; $PROC_{OEE}$ – processing time from the Overall Equipment Efficiency point of view (used in section 3.2.2).

To calculate $IDLE_{OEE}$ for the other four tool sets, it is necessary to have information about the number of setups. However, since the model uses the same setup rule, the number of setups will tend to zero as the load increases. Thus, we can ignore setup to calculate the approximate workload limit of the model. On the other hand, we use $PROC_{OEE}$ instead of $IDLE_{OEE}$ to build statistical models in section 3.2.2 for these four tool sets.

3.2 Two experimental scenarios (α_1 and α_2)

In previous studies, we used a detailed model to calculate the delays. This involved 2000 model runs for a given product mix (see Figure 2 scenario α_1 "Present"). However, this scenario is only suitable for research but not for practical use because it makes no sense to make a simplified model if we can afford 2000 runs of the detailed model for each product mix. Therefore, another scenario was developed α_2 ("Past + Future"), which assumes 12000 runs of the detailed model, but can be used for any product mix. This is accomplished by constructing and using the statistical forecasting model described below. The essence of the experiments in this paper is to compare the two scenarios (comparison Δ1 and Δ2 in Figure 2). The input data is the incoming lots' start rate and the simulation model structure. See formulas in subsection 3.2.2.

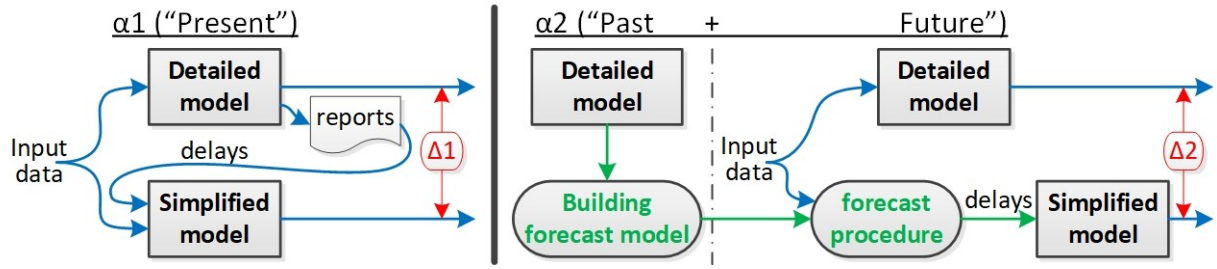

Figure 2: Two scenarios: α_1 ("Present") and α_2 ("Past + Future").

The statistical forecasting model is based on a similar concept to the operating curve, namely that for each tool set we can calculate delay = Cycle Time = f($IDLE_{OEE}$). However, difficulties have arisen in using this approach. When varying the model product mix, for several tool sets this dependence looks like a plane (w32, w27) while we expect a curve (w57, Figure 3). Obviously, there are some additional factors on which the Cycle Time (CT) depends. An investigation of these factors is beyond the scope of this paper. However, section 3.2.2 will show its results.

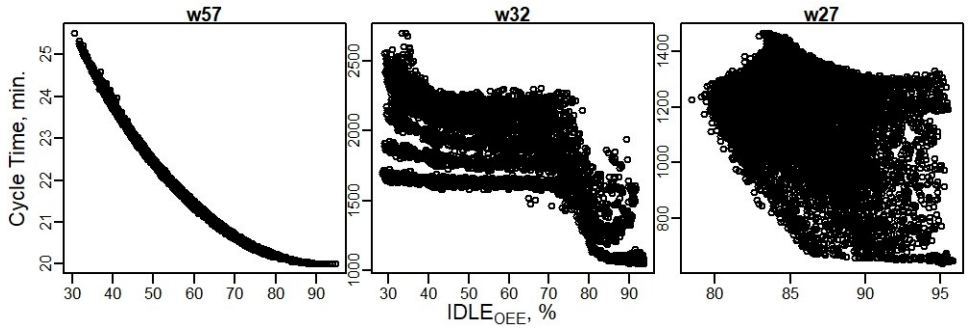

Figure 3: Simulation data CT($IDLE_{OEE}$).

The construction of a statistical forecasting model consists of the following steps: 1) construction of initiating data and 12000 runs of the detailed model in steady-state, 2) statistical approximation of the resulting output data, and 3) throughput evaluation for correct use under transient state conditions. The following three subsections briefly describe each of them.

3.2.1 Initiating data building

The initiating data is the input data for 12000 runs of the detailed simulation model in steady-state mode. The main goal is to obtain the widest variety of product mixes to cover all possible cases. After several preliminary experiments, it was decided to use 12 distributions to make this happen (Figure 4). Both individual distributions (e.g. a = Beta(0.15, 1)*65) and mixture distributions (e.g. l = 2/3* Beta(0.15, 1)*65 + 1/3*Uniform(0, 44.84)) were used. The distributions' parameters were chosen in order to ensure that the mean was approximately 8.5 (a priori estimation) and to provide the widest variety of product mixes. Figure 4 shows that in order to provide a product mix in which some products have very high demand and others very low, a mixture distribution had to be created. In this case, beta and uniform distributions were used for the right part to avoid getting "long tails" and thereby move the second part of the mixture distribution to the right as much as possible. Especially since the first three distributions already have "long tails".

Then, a product mix was generated based on each distribution. In the final initiating data, 1000 product mix combinations from each distribution were selected that provided $IDLE_{OEE}$ in the range (0%, 5%). In this way, the emphasis is on the high loading of the model. On the other hand, since in the MIMAC5 model, different tool sets correspond to different products, it is also possible that some tool sets are very heavily loaded while others are very weakly loaded. For this purpose, the distributions h, i, k, and l were used. Thus, we get the wide variety of loadings needed to build the statistical model for each tool set. However, more research is needed to determine whether this approach is exhaustive. For example, (Yang et al. 2007) suggested response surface modeling. This can be used as an alternative approach.

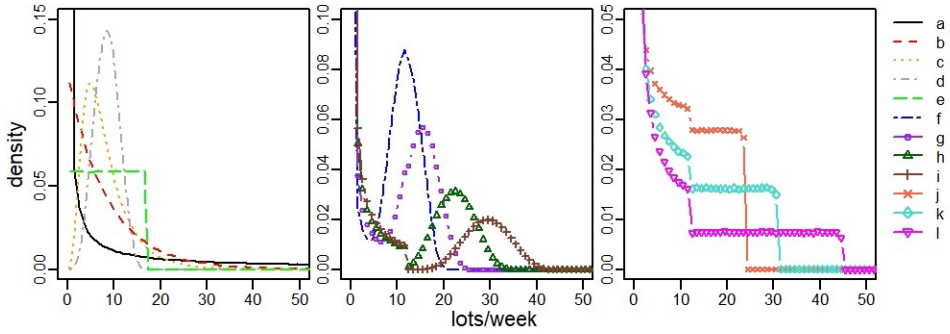

Figure 4: Initiating data. Probability density functions.

3.2.2 Statistical approximation

Since the Cycle Time (CT) data obtained using the initiating data cannot be described as a function of $IDLE_{OEE}$ (Figure 3), additional factors had to be introduced. After several experiments, the following expression was obtained (Figure 5):

$$CT = f(IDLE_{OEE} \text{ or } PROC_{OEE}, w.avg.RPT, batch\_entropy, setup\_entropy)$$

The formulas for $IDLE_{OEE}$, $PROC_{OEE}$, and *w.avg.RPT* are given above. $PROC_{OEE}$ was used only for 4 tool sets with setups. The variables *w.avg.RPT, batch_entropy,* and *setup_entropy* handle with the product mix variability estimation. Our primary interest is those tool sets that have process steps with different characteristics. Two different process steps belonging to the same tool set can differ in the following parameters: Raw Processing Time (RPT), Batch ID, and Setup ID (MIMAC datasets 1997). We consider the number of lots that pass through a given tool set and have a particular parameter value. Since different

process steps can belong to different process flows, the *w.avg_RPT* value gives us additional information about the product mix. For the other two variables, entropy was chosen because Batch ID is an identifier that tells us how the batch was formed but not the processing time. In other words, the batch information is a set of identifiers for lots passing through a given tool set. Similar logic was used for the Setup ID.

$$batch(\ or\ setup)\_entropy = -\sum_i^n \frac{lot\_num_i}{\sum_k^n lot\_num_k} \cdot log\left(\frac{lot\_num_i}{\sum_k^n lot\_num_n}\right)$$

where n is the number of Batch ID/Setup ID for the tool set, $lot\_num_{i\,(k)}$ is the number of lots for i^{th} (k^{th}) Batch ID/Setup ID for the given product mix.

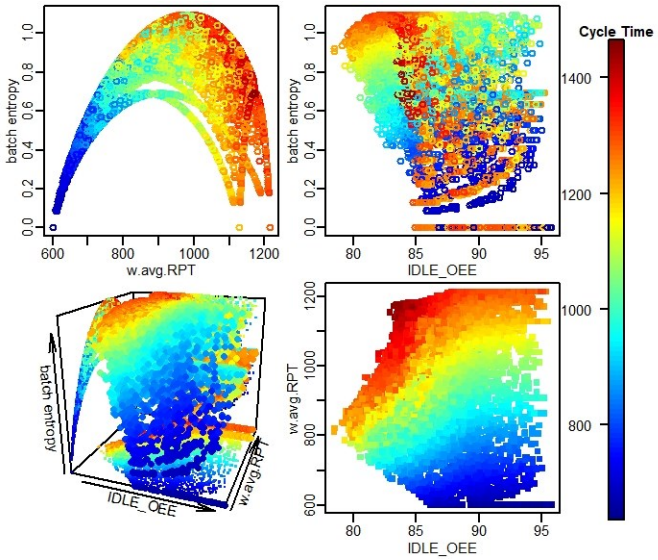

Figure 5: Tool set w27. 4D plot *CT = f(IDLE$_{OEE}$, w.avg.RPT, batch_entropy)*.

As a result of this approach, using the same data for the tool set w27 instead of the 2D plot (Figure 3), we can draw a 4D plot (Figure 5) in which CT is shown in color. Note that Figure 3 is a projection of the 4D plot of Figure 5 onto the coordinate axes (IDLE$_{OEE}$, CT). It can be seen that Figure 5 better describes the properties of CT. In particular, the projection in the plane (IDLE$_{OEE}$, w.avg.RPT) clearly shows the dependence of CT on these two parameters simultaneously. The R-package "mgcv" was used to build a statistical approximation based on the developed data model (Wood 2021).

3.2.3 Throughput evaluation

To use the statistical model built in the previous step, it is necessary to calculate the values of the factors using the formulas above. The most important input parameter is throughput (or *lot_num* in the formulas above). The problem is that for different tool sets, throughput at a given time (original TH) can be different from initial lot releases (initial TH). Note that the original throughput is derived from 2000 runs of the detailed model. Here we use it to illustrate the problem of evaluating the throughput. For example, for tool set w4, the detailed model shows a shift to the left for the initial TH relative to the original TH (Figure 6 a). On the other hand, there is no significant shift for the tool set w41 (Figure 6 b). This is explained by the different process steps associated with these tool sets. If the process steps are at the beginning of the process flow, there will be no shift.

Two approaches were developed to estimate throughput: 1) overall moving average (OMA); 2) processing step throughput adaptation (PSTA). The first approach consists of calculating for all process steps a simple moving average for initial lot release: $TH\_OMA = \sum_{i=n-k+1}^{n} TH_i/k$, where n is the current week number, k is the number of previous weeks. Then the throughput for all process steps belonging to

the given tool set is summed up. For simplicity, it is assumed that k is the same for all process steps. In the second approach for each process step, we calculate such values of weights ($\vec{w} = \{w_1, w_2, \dots, w_n\}$) and quantity of previous weeks l, which give the minimum value of the function: $th(\vec{w}, l) = \sum_{i=n-l+1}^{n} w_i \cdot TH_i / l \rightarrow min$. Then for each process step, a simple moving average is applied: $TH\_PSTA = \sum_{i=n-k+1}^{n} th(\vec{w}, l)_i / k$ and the parameter k is estimated. The meaning of the second step is explained further (Figure 7). In this paper, the calculation of $th(\vec{w}, l)$ was based on a detailed model since the main point was to test the principle feasibility of this approach. Nevertheless, it is possible to calculate $th(\vec{w}, l)$ based on a special statistical model, the development of which is beyond the scope of this paper.

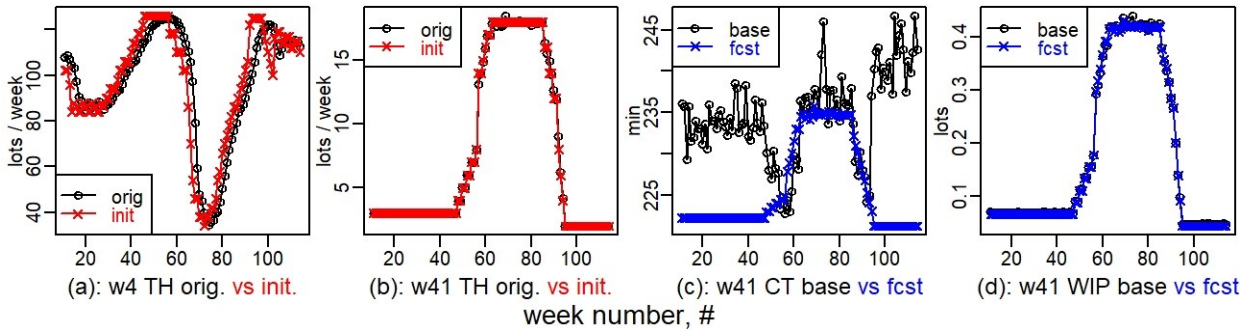

Figure 6: Throughput.

The purpose of the throughput evaluation is its use in the simulation model, i.e., it is not the throughput itself that is important, but its effect on the simulation model performance. Therefore, additional experiments were carried out with the detailed model to determine the parameter k (quantity of previous weeks), which would give the minimum value of the summed up for all tools set Mean Absolute Error (MAE) of Cycle Time (CT), Work-In-Process (WIP) and Throughput (TH). The results of the experiments are shown in Figure 7.

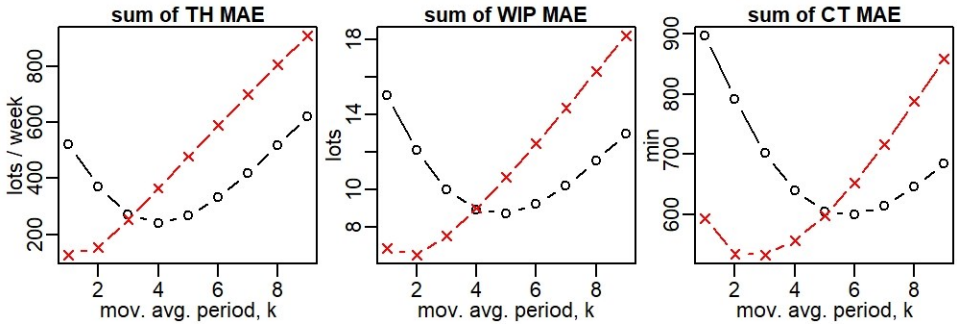

Figure 7: Throughput evaluation comparison: black (0) – OMA, red (x) – PSTA.

When analyzing the results of the experiments, it was observed that the tool set CT to a smaller degree than WIP reflects the accuracy of the model behavior as a whole. In particular, if only a few lots per week pass through a given tool set, they can have a significant variation in CT, which causes only a small variation in WIP (see Figure 6 b, c, and d). However, WIP takes into account not only the number of lots but also CT (Little's Law: WIP = CT*TH). Therefore, it was decided to use WIP as a criterion for the choice of parameter k. This decision leads to the following results: OMA(k = 5), and PSTA(k = 2). It can be seen that PSTA outperforms OMA in accuracy (Figure 7). Therefore, the PSTA(k = 2) method was used for the experiments described below.

4 EXPERIMENTS

The above-described statistical forecasting model was used in the simulation model simplification experiments. The main purpose of these experiments is to compare the results of the developed approach (α_2) with the baseline (α_1). An important issue is the method of comparison. In the paper (Stogniy and Scholl 2020), we discussed the use of accuracy measurements using a steady-state case. In this paper, we use the same accuracy measurements, but we need to make the necessary additions since we consider a changing product mix. The point is that for the steady-state, we calculate the values based on the entire runtime minus the warm-up period (i.e., 114 - 10 = 104 weeks). We call this approach "all". However, in the case of the changing product mix, we have time-varying data. Therefore, there is another approach to estimation: calculate the values based on the data calculated for each of the 104 weeks, and then aggregate them in order to obtain the average value of the index for the experiment (the "weekly average" or "w.avg" approach). Later in this section, we consider both approaches using two configurations as examples: $\{\zeta_2, \eta_1\}$ and $\{\zeta_9, \eta_1\}$. The first configuration is an example of close α_2 and α_1, the second of distant ones (see Table 1).

Figure 8 shows graphs of two accuracy measurements: Summarized Absolute Divergence (SAD) and Kolmogorov-Smirnov test (KS). It should be noted that we consider the output distributions of a lot CT. $SAD = \sum_x |PE(x) - PF(x)|$, where PE(x) is the probability density function (PDF) of sample 1 (detailed model), and PF(x) is the PDF of sample 2 (simplified model). $KS = \max|E(x) - F(x)|$, where E(x) is the empirical cumulative distribution function (ECDF) of sample 1 and F(x) is the ECDF of sample 2 (Stogniy and Scholl 2020).

Figure 8 illustrates the comparison of α_1 (black lines) and α_2 (red lines). The upper row of graphs is the "all" approach. The lower row is the "w.avg" approach. All four graphs for the same accuracy measurement (e.g., SAD: a, b, e, f) are drawn at the same scale. It is interesting to note that in all graphs, the red and black lines are quite close to each other, and the differences between the configuration $\{\zeta_9, \eta_1\}$ and $\{\zeta_2, \eta_1\}$ can also be easily observed. Based on the visual comparison, we can conclude that for the $\{\zeta_2, \eta_1\}$ configuration, the developed statistical forecasting model (α_2) works well in general.

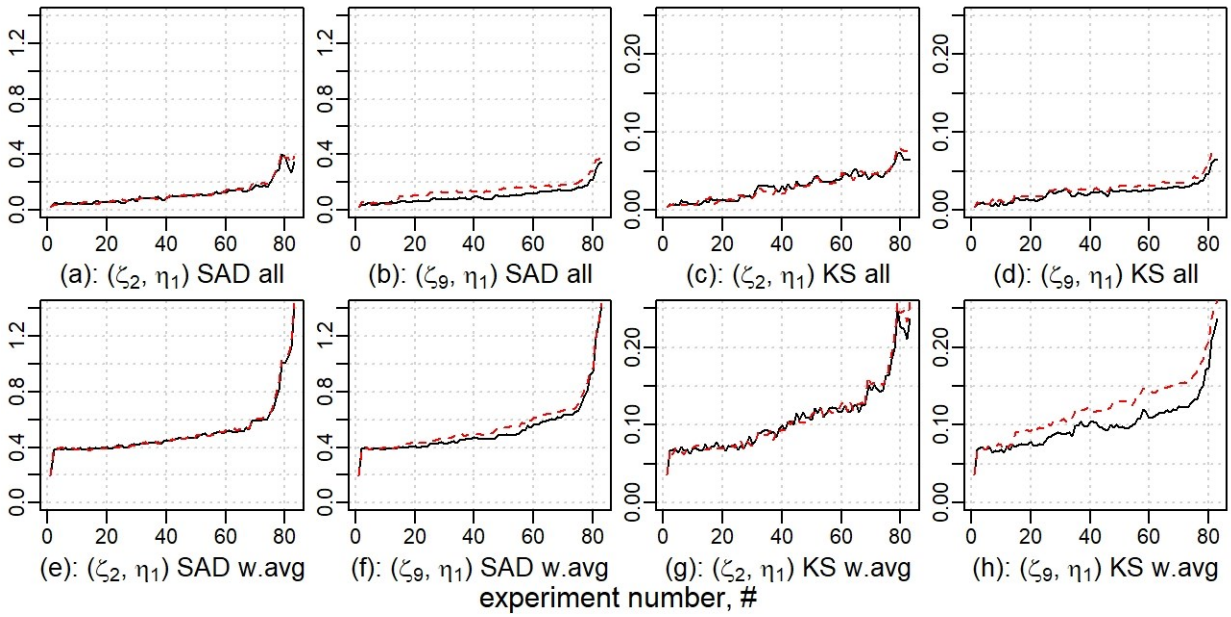

Figure 8: Comparison of α_1 (black solid) and α_2 (red dashed).

However, it is essential to consider and understand the differences between these graphs in order to choose the right way to compare α_1 and α_2. The first difference between the two calculation approaches: the

upper row of graphs ("all") has a much smaller scale and flatter curves than the lower row of graphs ("w.avg"). The second difference: "w.avg" has a rather high value of SAD and KS for small experiment numbers. To understand the reasons for the differences, it is necessary to consider the formulas for the accuracy measurements (see above) and to the input data for the calculations (Figure 9): probability density function (PDF) and empirical cumulative distribution function (ECDF).

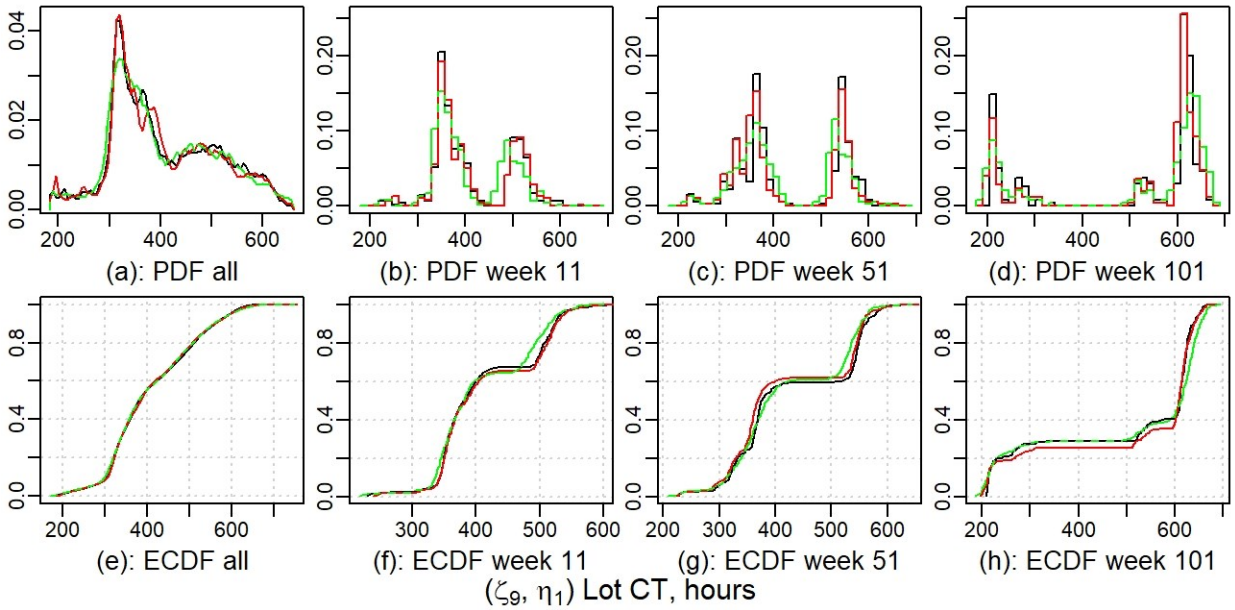

Figure 9: $\{\zeta_9, \eta_1\}$. Comparison of "all" and "w.avg".

Let us consider the first difference. Figure 9 shows the PDF and ECDF for the "all" approach (a and e) and for the three weeks (11, 51, and 101) of the "w.avg" approach (b-d and f-h). The lot CT distributions of the detailed model (baseline) are shown in green. Black and red show exp #60 for α_1 and α_2, respectively. Let us consider the top row (PDF). It can be seen from Figures 9b and 9c that the lot CT can vary greatly in different weeks: the peaks of the PDF curve shift to the right/left. This leads to a "blurring" of the total PDF (a) with regard to the PDF of each week (b-d). In addition, as we showed in the previous paper, there is a lot CT mean shift for large numbers of experiments (e.g., exp #60). This can be seen in the example of the right-hand PDF peak: week 11 – shift to the right, week 101 – small shift to the left. As a result, these shifts will compensate each other and will be almost invisible at PDF all (lot CT 400-700 hours). The situation is similar for ECDF. Here the difference between the forms of ECDF all and ECDF week 11, 51, and 101 can be seen even more clearly.

Let us consider the second difference: the high value of SAD and KS for small numbers of experiments in the "w.avg" approach compared to "all". Proceed in detail with the first two experiments: exp #1 and #2 (Figure 10). In this case, only one and two tool sets, respectively, were substituted for constant delays. As a result, the shift of the mean lot CT due to simplification cannot occur, but we see it for the right-hand peak PDF exp#2 week 11 (Figure 10c). In this case, the reasons for the shift are the butterfly effect and the small sample. The butterfly effect is a rearrangement of lots occurring even with a small change (Stogniy and Scholl 2020). It can be observed on the bottom row of scatter plots (base vs. exp). For exp #1, it appears insignificantly, and there is only a small scatter of points around the bisector (Figure 10e and 10f). However, for exp #2, the scatter of points is more significant (Figure 10g and 10h). Additionally, the small sample (only 5 seeds) leads to the fact that for exp #2 week 11, the lots with CT (450-600 hours) on average have higher CT than the same lots in the detailed model (base).

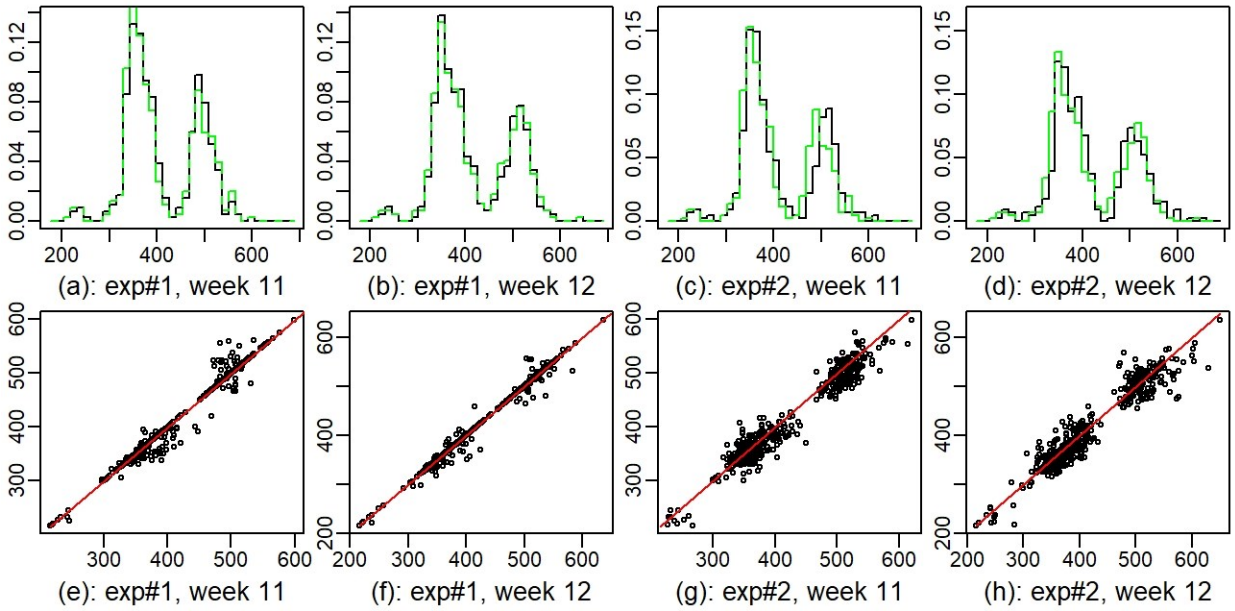

Figure 10: Comparison.

This work is part of a large exploratory study, and we could not have done more seeds as it would have significantly increased the overall experimentation time. However, according to our observations, this effect appears for all experiment numbers. Therefore, we can consider it as a systematic error, which we can neglect in our exploratory study. Moreover, in a more detailed study, the impact of this effect will weaken as the number of seeds increases. On the other hand, the first difference (Figure 9) does not depend on the number of seeds and has a significant impact on evaluating the simplified simulation model. Therefore, we conclude that the "w.avg" approach should be used to evaluate the results of the α_1 and α_2 approaches.

5 EXPERIMENTAL RESULTS

We illustrated two accuracy measurements above SAD and KS. However, in the paper (Stogniy and Scholl 2020), we considered four more: Anderson-Darling (AD) and Cramer-von Mises (CVM), Two Sample Test (DTS), and Wasserstein distance (WASS). Therefore, we use all six measures below (Table 1). We used the following formula to estimate the closeness of α_1 and α_2: $X\_index = 100 \cdot \left(\sum_i X_i^{\alpha 2} - \sum_i X_i^{\alpha 1}\right) / \sum_i X_i^{\alpha 1}$, where $X_i^{\alpha 1}(X_i^{\alpha 2})$ is the value of the X index in i^{th} experiment for α_1 (α_2).

The table shows that the values of the indices correlate with each other. However, the correlation is not 100%. The reason for this is the presence of a lot CT mean shift that occurs during simplification. SAD and KS are less sensitive to the shift than the other accuracy measurements (Stogniy and Scholl 2020). On the other hand, SAD is more sensitive to the butterfly effect, which leads to a high denominator value in the formula above, and as a result, less scatter for the *X_index*. Therefore, we believe that KS is the most appropriate criterion for evaluation. Those configurations for which *KS_index* ≤ 3% are highlighted in green in the table, and those for which *KS_value* >10% are highlighted in red. This table shows that the configurations associated with ζ_2, ζ_3, and ζ_6 show sufficient closeness of α_1 and α_2. On the other hand, the configurations ζ_9 and ζ_{10} lead to poor results. The reason seems to be the different manifestations of the lot CT mean shift for the different configurations. However, this issue requires further investigation. In general, we can admit that the developed statistical model within the α_2 scenario works satisfactorily. This fact brings us closer to the use of simplified models in a real industry.

Table 1: Configuration comparison, %.

ζ	η	SAD	KS	AD	CVM	DTS	WASS	ζ	η	SAD	KS	AD	CVM	DTS	WASS
1	**1**	1.4	6.6	16.6	19.8	4.6	6.5	**6**	**1**	0.4	2.6	9.5	9	2.7	2.8
1	**3**	1.7	6.3	13.9	17.2	3.2	4.8	**6**	**3**	1.1	3	8.5	8.3	2.4	2.8
1f	**1**	1	3.2	10.7	11	2.9	3.5	**6f**	**1**	0.8	2.5	7.8	7.2	2.1	2.3
1f	**3**	0.8	2.7	7.5	8.1	1.5	2.1	**6f**	**3**	0.7	2.1	7	6.6	2.5	2.6
2	**1**	0.8	1.7	8.1	6.5	2.6	2.5	**7**	**1**	1.3	3.1	10.8	9.6	3.3	3.8
2	**3**	1	2.7	10.8	10.2	2.8	2.8	**7**	**3**	0.8	3.1	10.6	9.6	2.2	2.6
2f	**1**	1	2.1	7.4	5.8	3.3	3.3	**7f**	**1**	1.3	3.1	10.1	8.6	3.3	3.5
2f	**3**	1	2	9	6.9	2.4	2.4	**7f**	**3**	1	2.8	10.3	8.8	2.9	3
3	**1**	0.8	2.7	9.2	8.1	3.7	4.2	**8**	**1**	2.3	7.7	22.1	25.4	7.4	9.3
3	**3**	0.3	0.5	-3.2	-3.8	-0.7	-0.2	**8**	**3**	1.1	3.8	7.8	12.2	1.7	3.6
3f	**1**	0.9	2.9	9.1	8.8	3.7	4.1	**8f**	**1**	2.9	9.7	26.8	29.9	9.1	10.9
3f	**3**	0.4	1.3	-0.7	-0.8	0.1	0.7	**8f**	**3**	1.5	4.9	10.2	13.2	2.5	4.3
4	**1**	1.3	3.6	14.2	12.5	3.6	4.2	**9**	**1**	5.8	21.1	46	56	15.4	20.4
4	**3**	0.1	0.4	-0.9	-1	-1	-0.6	**9**	**3**	5.9	20.4	47.1	57.6	16.1	20.5
4f	**1**	1.3	4.4	15.1	13.3	5	5.5	**9f**	**1**	5.4	20.8	43.4	52.7	15.4	20.2
4f	**3**	0.2	0.1	0.8	-0.8	-1.3	-0.9	**9f**	**3**	5.7	20.4	46.2	55.1	16.5	21
5	**1**	1.4	5.5	19	18.7	5.7	6.3	**10**	**1**	3.3	13	27.4	33.5	12.6	15.7
5	**3**	2.1	7.4	23.4	24.8	7.6	8.8	**10**	**3**	2.6	8.4	12.2	18.8	4.7	8.7
5f	**1**	1.8	5.8	19.9	20	5.7	6.5	**10f**	**1**	3.2	12.6	26.7	32.8	12	15
5f	**3**	2.3	6.9	22.6	22.7	6.9	7.7	**10f**	**3**	2.1	7.3	11.4	17.6	4.1	7.8

6 CONCLUSIONS

In this paper, we considered using a simplified simulation model under near-real-world conditions, i. e., when there is no information about the behavior of the detailed model in the future (scenario α_2). For this purpose, we created and described a forecasting model that can be used for any product mix. The key components of the forecasting model are initiating data, statistical approximation, and throughput evaluation. Using the proposed statistical models showed promising results in general. We also refined the way of using the previously developed accuracy measurements for the changing product mix scenario and described the effects that affect the accuracy of the simplified model: the lot cycle time mean shift and butterfly effect.

Based on our experiments, we can conclude that applying utilization-based (ζ_2 and ζ_3) and queue length-based (ζ_6) sieve functions leads to acceptable accuracy using the forecasting model (α_2) for our case, the MIMAC 5 model. However, because of the lot cycle time mean shift influence on the accuracy of the simplified model, we cannot claim that for other simulation models, the same sieve functions will show good performance. On the other hand, we can almost certainly state that these effects (lot cycle time mean shift and butterfly effect) will appear for other models as well. The techniques and methods described in this paper will allow future researchers to detect this fact in time.

In future research, we will consider the application of the above approaches to the Critical Ratio dispatching rule. We will also perform a more in-depth analysis of using representative process flows and artificial process flows. Additionally, we plan to conduct optimization experiments with simplified models. Eventually, these approaches will be implemented in a real Infineon simulation model.

ACKNOWLEDGMENTS

A part of the work has been performed in the project iDev40. The iDev40 project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 783163. The JU receives support from the

European Union's Horizon 2020 research and innovation programme. It is co-funded by the consortium members, grants from Austria, Germany, Belgium, Italy, Spain, and Romania.

REFERENCES

Albey, E., U. Bilge and R. Uzsoy (2017). "Multi-Dimensional Clearing Functions for Aggregate Capacity Modelling in Multi-Stage Production Systems." *International Journal of Production Research* 55(14): 4164-4179.

Beeg, T. 2004. "Wafer fab cycle forecast under changing loading situations". In *2004 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*. 339-343.

Duarte, B. M., J. W. Fowler, K. Knutson, E. Gel, and D. Shunk. 2007. "A compact abstraction of manufacturing nodes in a supply network". *International Journal of Simulation and Process Modelling*, 3(3), 115-126.

Ewen, H., L. Mönch, H. Ehm, T. Ponsignon, J.W. Fowler, and L. Forstner. 2017. "A Testbed for Simulating Semiconductor Supply Chains". *IEEE Transactions on Semiconductor Manufacturing* 30(3): 293-305.

Huang, P. B., J. C. Chen, C. W. Yu, and C. C. Chen. 2016. "Integration of simulation and neural network in forecasting the throughput for TFT-LCD colour filter fabs". *International Journal of Computer Integrated Manufacturing*, 29(3), 298-308.

Hung, Y. F. and R. C. Leachman. 1999. "Reduced Simulation Models of Wafer Fabrication Facilities". *International Journal of Production Research* 37(12): 2685-2701.

Jain, S., B. P. Gan, C. C. Lim, and Y. H. Low. 2000. "Bottleneck Based Modelling of Semiconductor Supply Chains". In *Proceedings of the International Conference on Modeling and Analysis of Semiconductor Manufacturing*.

MIMAC Datasets. 1997. http://p2schedgen.fernuni-hagen.de/index.php?id=simulation&L=1, accessed 29th March 2021.

Missbauer, H. and R. Uzsoy (2020). Production Planning with Capacitated Resources and Congestion. New York, NY, Springer.

Mönch L., R. Uzsoy and J.W. Fowler. 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.

R Core Team 2019. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/, accessed 29th March 2021.

SEMI E79-0200. 2000. Standard for Definition and Measurement of Equipment Productivity.

Sprenger, R., and O. Rose. 2010. "On the Simplification of Semiconductor Wafer Factory Simulation Models". In *Conceptual Modeling for Discrete-Event Simulation*, edited by S. Robinson, R. Brooks, K. Kotiadis, and D.J. van der Zee, 451-470. Boca Raton: CRC Press.

Stogniy I. and W. Scholl. 2020. Using accuracy measurements to evaluate simulation model simplification. In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. Bae, B. Feng, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1825 – 1836. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Van der Zee, D. J. 2019. "Model Simplification in Manufacturing Simulation – Review and Framework". *Computers & Industrial Engineering*, 127: 1056-1067.

Völker S. and P. Gmilkowsky. 2003. "Reduced Discrete-Event Simulation Models for Medium-Term Production Scheduling". *Systems Analysis Modeling Simulation* 43(7): 867-883.

Wang, J., P. Zheng, and J. Zhang. 2020. "Big data analytics for cycle time related feature selection in the semiconductor wafer fabrication system". *Computers & Industrial Engineering*, 143: 1-11.

Wood, S. (2021). Package 'mgcv'. R package version 1.8-34. URL https://CRAN.R-project.org/package=mgcv, accessed 29th March 2021.

Yang, F., B. Ankenman, and B.L. Nelson. 2007. "Efficient generation of cycle time-throughput curves through simulation and metamodeling". *Naval Research Logistics*, 54(1): 78-93.

AUTHOR BIOGRAPHIES

IGOR STOGNIY is a PhD-student at the Technische Universität Dresden (Germany). He received his M.Sc in Automation and Control in 2007 from Bauman Moscow State Technical University (Russia). Since 2012 he has been working at the TU Dresden as a scientific researcher in the area of simulation. His email address is igor.stogniy@tu-dresden.de

WOLFGANG SCHOLL works as a Senior Staff Expert for modeling and simulation for Infineon Technologies in Dresden (Germany). He studied physics at the Technical University of Chemnitz (Germany) and graduated in solid-state physics in 1984. Since 2003 he has been responsible for fab simulation in Infineon Technologies. He supervises development and roll-out projects and is also a member of the Supply Chain Simulation community. His email address is wolfgang.scholl@infineon.com.

HANS EHM is Lead Principal Supply Chain heading the Corporate Supply Chain Engineering Innovation department at Infineon Technologies. He has a Master of Science in Mechanical Engineering from Oregon State University. His interest is on Supply Chain Innovation with focus on capacity and demand planning, deep learrning, artificial intelligence, simulation, and block chain applications for traceability. His email address is hans.ehm@infineon.com.