Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

METRICS TO EVALUATE SIMPLIFICATION

Igor Stogniy

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

Wolfgang Scholl

Infineon Technologies Dresden
Königsbrücker Straße 180
Dresden, 01099, GERMANY

ABSTRACT

Infineon Technologies Dresden has a long-term experience in discrete event simulation which is used for an optimized production planning for its full automated front end manufacturing lines. There are needs to reduce maintenance effort; to increase transparency for the plausibility check facilitation; to improve flexibility for the fast simulation of scenarios with focus on qualitative statements. Less detailed models will be utilized where components could be omitted. We considered a simplification of the process flows through operation substitution for constant delays. The main idea of this paper is to consider the metrics used to evaluate simplification. It is shown that standard simulation metrics (mean absolute error, correlation coefficient, autocorrelation function, t-test) produce rather poor performance. It is suggested instead to use metrics based on lot cycle time distributions (e.g. goodness-of-fit tests). Nine types of simplification sieve functions were analyzed. The analysis is based on the MIMAC dataset 5 model.

1 INTRODUCTION

Van der Zee (2019) presented an overview of the simulation model simplification and noticed that simplification “is still very much a green field”. Numerous papers have been written around this theme. Unfortunately, some of them describe evaluation metrics which are only suitable for the rather simple examples which the authors considered in their works. More appropriate methods do exist, e.g. goodness-of-fit tests (Law 2015), but they are not as popular. This is possibly due to historical traditions in simulation, where the main focus is placed on precise estimation of the mean and quantile values of the simulation model results in comparison with reality (Alexopoulos and Kelton 2017), but not on the comparison of the dozens of simulation models with each other.

As a continuation of the work which we presented earlier (Stogniy and Scholl 2019), the current paper makes the following contributions: 1) gradual simplification with substitution of a one tool set per step (this allows oscillations in the metrics to be seen); 2) the consideration not only of machine tool delays, but also operation based delays (providing an improved simplified model); 3) considerations of 9 (not one and a half) sieve functions with a focus on standard simulation metrics (mean absolute error, correlation coefficient, autocorrelation function, t-test) as opposed to simply comparing the results of sieve functions, and we found the metrics not suitable for simplification tasks, in contrast to some findings in the literature; and 4) a dramatically increased data volume for which R (R Core Team 2019) was used for metrics calculation, as opposed to the electronic spreadsheets used previously.

As a basis for the research, the well-known MIMAC Datasets (1997) were used. MIMAC dataset 5 was chosen because it has the largest quantity of process flows. For the fab simulation model, AutoSched AP from Applied Material, version 11.5, was used.

This paper is organized as follows. Several ideas from related works are presented in Section 2. The designs of the experiments, including sieve function, delay types, and metrics description, are discussed in Section 3. The diagrams, including metrics and their critiques, are described in Section 4. Section 5 includes summarized experimental results for sieve function comparison.

2 RELATED WORK

This research was inspired mostly by two papers: Johnson et al. (2005) and Völker and Gmilkowsky (2003). Following Johnson et al. (2005), gradual simplification was implemented in this paper, but with a much more detailed model (MIMAC dataset 5 and not tandem-10 M/M/1 or M/M/10 model). Johnson et al. (2005) used correlation and autocorrelation to estimate the accuracy of simplified models. In this paper it was determined that these estimations would not be sufficient. Völker and Gmilkowsky (2003) was found to be the most detailed paper in the field of simplification. In that paper, the authors used three sieve functions to choose the operations for substitution. Our research uses nine sieve functions, each with two variations. The focus was put on machine tools and not on the operation substitution, as this does not require capacity reservations. However, information about each particular operation was also used for the calculation of operation-based delays. The authors also used an estimation based on lot cycle time mean absolute error (MAE) and found "pathological" configurations when the error is not monotonous. We also found such configurations and assume them to be "logical". Moreover, we found MAE was inadequate as an estimation tool for simplified models. Similar to the authors, we used an operation ratio to quantify the degree of simplification.

Another consequential paper is that of Rose (1999), from which several design features were adopted: using First In First Out (FIFO) and Critical Ratio (CR) dispatching rules, considering lot cycle time distribution as an accuracy criterion, and looking into bottleneck problems. Rose (1999) did not present formal criteria to estimate differences in lot cycle distributions, but used them as illustrations. In a previous paper (Stogniy and Scholl 2019), we suggested using a Kolmogorov-Smirnov test and summarized absolute divergence (SAD). Within the review process, out attention was directed to a similar SAD criterion in (Ewen et al. 2017). In this paper we considered these and other metrics. It is also necessary to emphasize that one of the popular metrics, t-test (see for example Piplani and Puah (2004)), is not suitable for the simplification in our case. A literature review of papers on bottlenecks produced two helpful ideas: a sieve function based on active period method (Roser and Nakano 2015) and using standard deviation to estimate sieve function fluctuation.

3 DESIGN OF EXPERIMENTS

For this research a special automated experimental environment was developed. It allowed for 4,000 experiments with detailed models to calculate warm up period and delays (2000 seeds x2: FIFO and CR dispatching rules). Subsequently, we made 44,820 automated experiments for KPI calculations: 3 system configurations: ε_1 – FIFO, ε_2 – CR, ε_{2b} – CR with consolidated delays at the beginning; 9 sieve functions; 2 variants of sieve function fluctuation (with and without); 2 types of delays (operation and machine tool based); and 83 tool sets substitutions (5 seeds for each substitution).

We consider a steady state simulation. Each simulation run is 114 weeks. We used Welch's procedure to determine the warm up period (Law 2015). The warm up was set to 10 weeks. We consider tool set substitution for constant but not random delays. In (Stogniy and Scholl 2019) we explained this choice. In this paper we used distributed delays (ε_1 and ε_2). However, because the previous research showed a valuable difference for CR rules if we calculate a consolidated delay for each process flow and put it at the beginning, we considered this system configuration (ε_{2b}) as well.

3.1 Sieve functions ζ

For tool sets substitution we used some heuristics, which we called sieve functions following (Völker and Gmilkowsky 2003). We created an order of tool sets according to a particular sieve function and substitute a tool set one by one. Usually in simplification papers the term "bottleneck" is used. We prefer to use the term "tool importance" to emphasize that we are not looking for a real bottleneck, but just trying to find an important tool set when considering substitution.

We used the following model statistics based on weekly standard model reports to build sieve functions: *IDLE%/IDLE#* – the percent of time/the number of times a machine tool entered the idle state;

$PROC\%$ – the percent of time a machine tool 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 machine tool (queue time); QL_{AVG} – the average number of pieces in front of the machine tool (queue length); PT_{AVG} – the average of the lot processing time for the machine tool; CT_{AVG} – the average lot cycle time for the machine tool ($CT_{AVG} = PT_{AVG} + QT_{AVG}$); and CT_{SD} – standard deviation of the cycle time for the machine tool ($CT^2_{SD} = PT^2_{SD} + QT^2_{SD}$).

We used the following sieve functions: $\zeta_1 = IDLE\%$; $\zeta_2 = IDLE\% + PROC\% - PROC\%(BS_{AVG} / BS_{MAX})$; $\zeta_3 = (100 - IDLE\%) / IDLE\#$; $\zeta_4 = QT_{AVG}$; $\zeta_5 = QT_{AVG} / PT_{AVG}$; $\zeta_6 = QL_{AVG}$; $\zeta_7 = QL_{AVG} / BS_{MAX}$; $\zeta_8 = CT_{SD}^{total}$; and $\zeta_9 = CT_{SD}^{total} / CT_{AVG}^{total}$. All of ζ (except ζ_3) were calculated based on weekly report data of 2,000 independent experiments with detailed models. ζ_3 was calculated based on 104-week period reports of the 2,000 experiments. This was done to reduce the cases where $IDLE\# = 0$. In the case of 104-week period reports, we had $IDLE\# = 0$ only for one machine tool in about 500 of 2,000 experiments. In this case we substituted it with $IDLE\# = 1$ for the calculation of ζ_3. To calculate CT_{SD}^{total} (ζ_8), we used the following formula (the law of total variance (Fewster 2014)):

$$\mathrm{CT}_{SD}^{total} = \sqrt{\frac{\sum_{i=1}^{n}(CT_{SD})_i^2 \cdot p_i}{\sum_{i=1}^{n} p_i} + \frac{\sum_{i=1}^{n}\left((CT_{AVG})_i - CT_{AVG}^{total}\right)^2 \cdot p_i}{\sum_{i=1}^{n} p_i}},$$

where i is the report period (a week), $n = 104$ weeks, and p_i is the number of pieces which finished processing at the machine tool during the i^{th} report period.

Several papers about bottlenecks (e.g. Roser and Nakano (2015)) mentioned that taking momentary bottlenecks into account is important, and we tried to implement this idea into our research. Therefore we considered sieve function fluctuations. As long as we calculated ζ based on large amounts of data, we had a distribution of its value and calculated mean value (μ) and standard deviation (σ). We considered two variants of sieve function: $\zeta_i = \mu$ and $\zeta_{if} = \mu \pm \sigma$ (where i = 1, 2, …, 9 and f – fluctuation). To understand the difference between the two variants and which sign ("+" or "–") we should use in the second variant, let us consider a conditional example (Figure 1). There are three machine tools (A, B, and C) with the following case to consider: "tool importance" → max, when ζ → min. This means that we need to create an ascending order of the machine tools. If we take into account only mean values (μ), then we get an order B<A<C. But if we take mean and standard deviation values, we should use μ – σ to obtain a conservative estimation. Therefore we have A<B<C. If we consider another case: "tool importance" → max, when ζ → max, then we should take μ + σ.

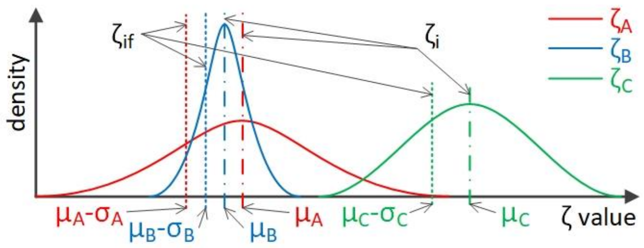

Figure 1: A conditional example of ζ fluctuation. Two variants: ζ_i and ζ_{if}.

3.2 Delay types η

We consider two types of delays: operation (η_1) and machine tool (η_2) based. The delays are calculated as averages based on 104 weekly reports of 2,000 independent experiments with detailed models. The first type is calculated based on operation reports and the second type based on tool set reports (Figure 2). A delay for a particular week is equal to the sum of average processing and average queueing time for an operation or a tool set. For system configurations FIFO(ε_1) and CR (ε_2) we used distributed delays: each delay substitutes each operation exactly at the place where this operation appears in the process flow. For the case CRb (ε_{2b}) we calculated the sum of all the delays for a process flow and put it at the beginning of the process flow.

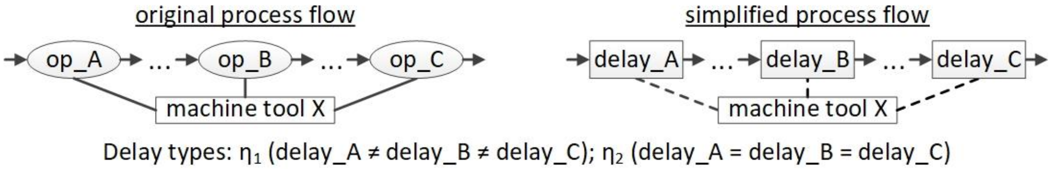

Figure 2: Delay types: η_1 – operation based; η_2 – machine tool based.

Thus for each of the nine sieve functions we have four experiment configurations: (ζ_i, η_1), (ζ_i, η_2), (ζ_{if}, η_1), and (ζ_{if}, η_2). Altogether we have 36 experiment configurations for each of the three system configurations (ε_1 – FIFO, ε_2 – CR, ε_{2b} – CRb). Later in this paper we use a triad (ε, ζ, η) to label a particular experiment configuration.

3.3 Metrics

In this paper we used the following simulation run data metrics: simulation time, operation ratio (OP_r = 100 * substituted operations/all operations); Work-In-Process ratio (WIP_r =100 * WIP of substituted tool sets/WIP of all tool sets); Work-In-Process in the queue ratio (WIPq_r = 100 * WIP of the queue in front of substituted tool sets/ WIP of the queue in front of all tool sets). We calculated WIP values based on detailed models. We also used four typical metrics from simplification papers: lot cycle time (CT) mean absolute error (MAE), correlation coefficient, autocorrelation function, and t-test.

After several experiments and unexpected results (see "pathological" in the Section 4.1), we found the following metrics to be useful based on lot CT distribution (Dowd 2018): Kolmogorov-Smirnov (KS), Anderson-Darling (AD) and Cramer-von Mises (CVM), Two Sample Test (DTS), Wasserstein distance (WASS), and Summarized Absolute Divergence (SAD). KS and SAD were introduced in our previous paper (Stogniy and Scholl 2019). There we used Excel to calculate them, while in this paper we used R to calculate the metrics. For the KS test we used the standard function from the R-package "stats" and for SAD we wrote our own R-code. There were difficulties in implementing Anderson-Darling and Cramer-von Mises tests with several R-packages (some functions have worked too long and others have not worked at all for our data) until we found a package "twosamples". The package showed good performance. We compared the KS test from packages "stats" and "twosamples" and found the results to be in agreeance. Therefore we assumed that other statistics are also calculated properly in this package and we used it for our research. The names DTS and WASS seem incorrect, but we used them just as labels. In the end, it is the formulas behind the labels that matter. The formulas (Table 1) were taken from the package description (Dowd 2018).

Table 1: Metrics.

Designation	R package / function	Formula
KS	stats/ks.test	$\max\lvert E(x) - F(x)\rvert$
AD	twosamples / ad_stat	$\sum_x \left(E(x) - F(x)\right)^2 / [G(x)(1 - G(x))]$
CVM	twosamples / cvm_stat	$\sum_x \left(E(x) - F(x)\right)^2$
DTS	twosamples / dts_stat	$\sum_x \lvert E(x) - F(x)\rvert / [G(x)(1 - G(x))]$
WASS	twosamples / wass_stat	$\sum_x \lvert E(x) - F(x)\rvert$
SAD	own code	$\sum_x \lvert PE(x) - PF(x)\rvert$

Table 1 displays the designation, source, and formulas used for calculating the statistics. In the formulas: E(x) is the empirical cumulative distribution function (ECDF) of sample 1 (detailed model); F(x) is the ECDF of sample 2 (simplified model); G(x) is the ECDF of the joint sample; PE(x) is the probability density function (PDF) of sample 1, and PF(x) is the PDF of sample 2. Because the R-package "twosamples" uses randomization to create p-values (bootstrapping), it takes significant time to produce meaningful numbers. Therefore, we did not calculate p-values for AD, CVM, DTS, and WASS. For SAD it is not possible to calculate a p-value, because it is not a statistic in a classical meaning, but just a metric which we found suitable.

4 EXPERIMENTS

Five independent experiments with different seeds were carried out for each tool set substitution. After that, an average of output parameters (simulation time, correlation coefficient, etc.) were calculated to compare different experiments. Figure 3a illustrates the calculation for simulation time. The calculations were carried out on a server which is used for various tasks simultaneously. We assume that fluctuations in simulation time were caused by fluctuations in server load by other tasks.

Figure 3b demonstrates relations between simulation time, operation ratio, and WIP ratios (here and later, all curves represent the average of the five seeds). The resulting curves appear different for different experiment configurations. Often they are relatively close to each other, but for some cases, such as (ε_1, ζ_8, η_1), only operation ratio repeats the simulation time curve convexity. Moreover, Figure 3b shows that simulation time is not a monotonically decreasing function, in contrast to an operation ratio. Therefore we used operation ratio as a measurement of simplification to compare different experiment configurations at the end of this paper (Figure 9). On the other hand, using the experiment number on the x-axis makes the diagrams more intuitive and understandable. Therefore, we used this x-axis for the following figures.

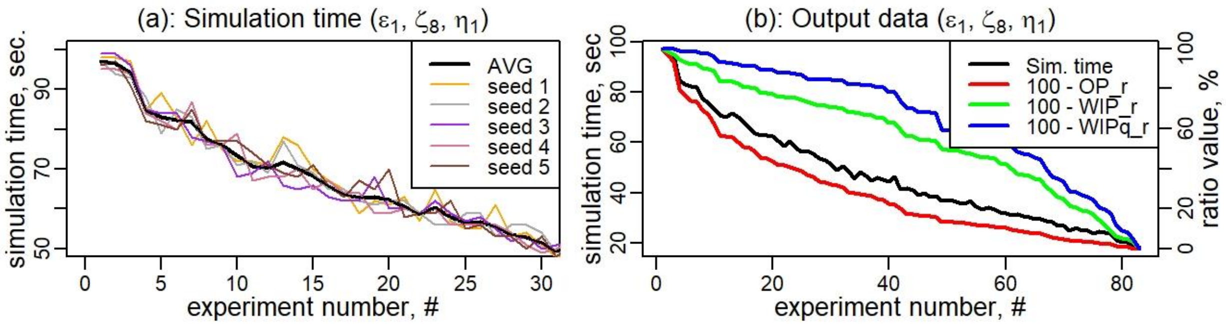

Figure 3: Simulation output.

4.1 Typical metrics

A common metric of model accuracy is lot cycle time (CT) mean absolute error (MAE). Völker and Gmilkowsky (2003) used a metric based on MAE and found some configurations which they called "pathological", where an error does not grow monotonously, but increases at the beginning and decreases at the end. For example, (ε_1, ζ_1, η_1) from our experiments: MAE increases until experiment number 60 and falls after 70 (Figure 4a). At first we assumed that this behavior was caused by a shift of the overall lot CT mean. An analysis of this idea found a confirmation (Figure 4b). But then we found another example (ε_1, ζ_7, η_1) where we had similar overall mean shift. Therefore, we also looked at lot CT standard deviation (SD) (Figure 4b). An average SD based on each process flow SD was calculated. The average SD decreases in both cases, but in the first case more. Figure 4c and 4d illustrate this behavior: the exp#70 distribution for (ε_1, ζ_1, η_1) is much more narrow than for (ε_1, ζ_7, η_1), but there is a significant shift to the left at the right peak for both cases. We could conclude that "pathological" behavior exists when we have both lot CT mean shift and a strong decrease of average SD (ε_1, ζ_1, η_1). If there is only the shift and a mild decrease of average SD, then no "pathology" is observed (ε_1, ζ_7, η_1). There were several other examples of "pathological" but less expressed behavior: (ε_1, ζ_3, η_1), (ε_1, ζ_4, η_1), and (ε_1, ζ_5, η_1).

Lot CT mean shift and decrease of average SD were a result of substitution for constant delays. If the SD reduction is logical, due to the use of constant and not random delays, the shifting of the lot CT mean indicates that the calculation of constant delays as a sum of averaged processing and queueing time is not necessarily correct. What is more interesting is that the shift has negative ((ε_1, ζ_1, η_1) exp# 40-70) and positive ((ε_1, ζ_1, η_1) exp# 70-80) values. Identifying a more accurate way to calculate the delays would be a worthwhile endeavor for future research.

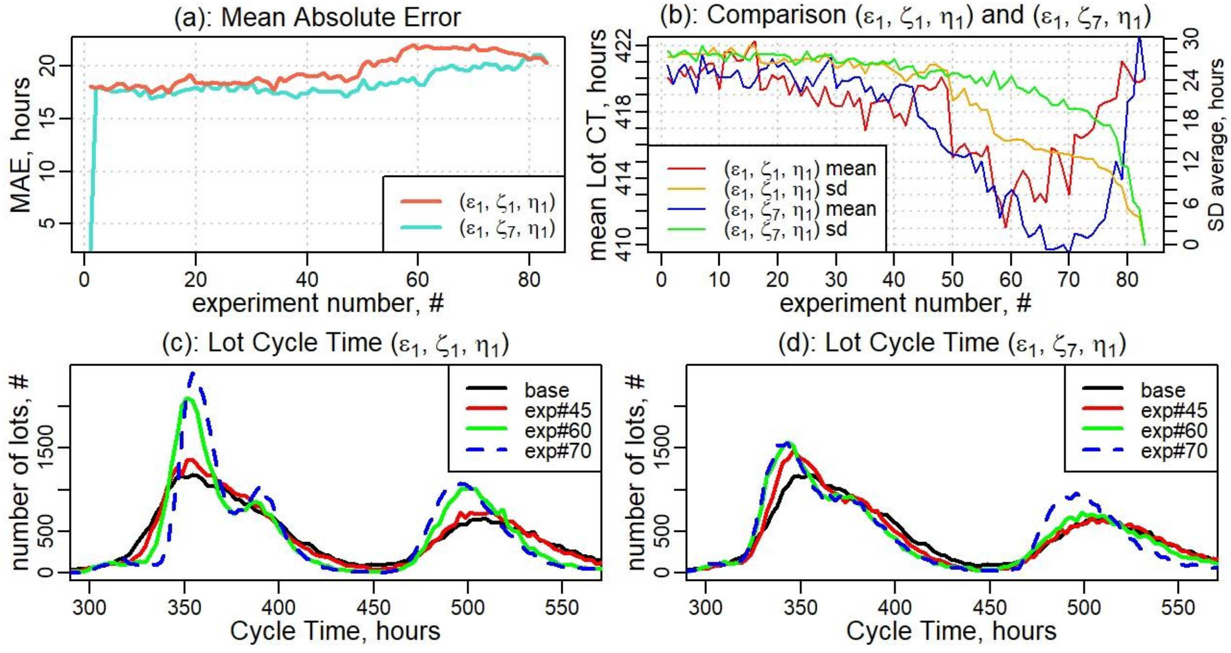

Figure 4: (a) MAE; (b) mean lot CT and average SD; (c) and (d) lot CT distributions.

Another interesting observation about MAE is that, for some configurations (e.g (ε_1, ζ_1, η_1) see Figure 4a), the MAE value is high (18.08) even in the first experiment, and there is minimal change (20.3) by the last experiment. This suggests that the first substitution of a tool set with IDLE% = 97.14 ((ε_1, ζ_1, η_1) exp#1) leads to significant changes in MAE (from 0 to 18.08), but further substitutions incur only small changes (from 18.08 to 20.3). For other configurations (e.g (ε_1, ζ_7, η_1), this happens after the second substitution. Two possible reasons for this exist: 1) MAE calculation based on every single lot; 2) even small changes in the production system could lead to big lot rearrangements. We could call it a "butterfly effect". The scatter plot in Figure 5b illustrates this phenomenon for the exp#1 ((ε_1, ζ_1, η_1).

A second typical measurement for model accuracy is correlation coefficient. Figure 5 demonstrates that the correlation is a less effective measurement for the simplified model. We can see that for the configuration (ε1, ζ_7, η_1) the correlation oscillates from exp#2 to exp#77 between 0.95 and 0.96 (Figure 5a). Moreover, for the configuration (ε_1, ζ_1, η_1) we have again "pathological" behavior. We could assume that indeed the correlation coefficient is not sensitive to the simplification. Scatter plots in Figure 5 illustrate the reason for this: despite the fact that we have a point cloud in the scatter plot, the cloud is distributed along the bisector. Moreover, the cloud narrows to parallel lines as the experiment number increases (compare b with e and c with f). Figure 5d shows the scatter plot for the extremely reduced model (exp#83). Note that the correlation coefficient is 0.947 in this case.

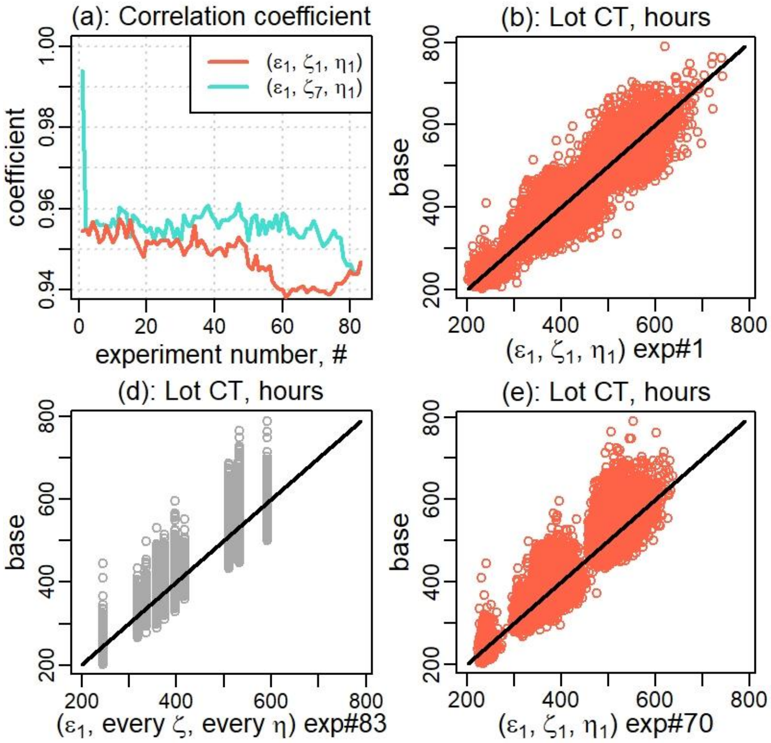

Figure 5: Correlation coefficient and scatter plots.

The third typical criteria to determine the accuracy of simplified models is the value of autocorrelation function. Rose (1999) and Johnson et al. (2005) noted that autocorrelation should decrease with simplification. This is true if we consider the very simplified model of exp#80, where all except 3 tool sets were substituted for delays (see Figure 6).

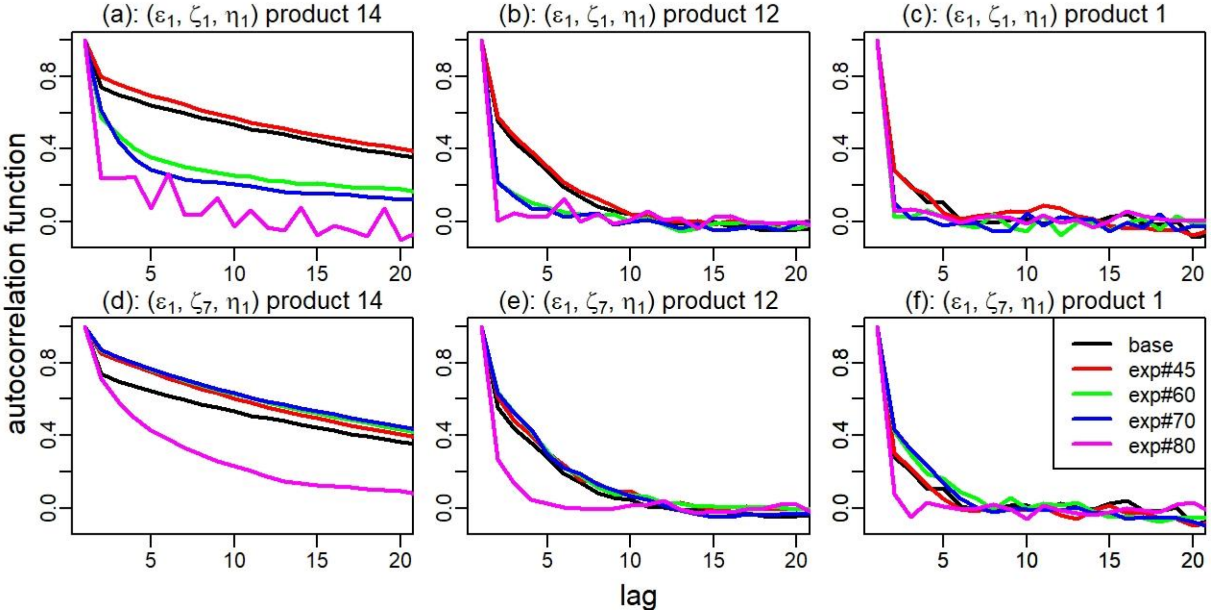

Figure 6: Autocorrelation function (base case, experiments: 45, 60, 70, 80).

But attempting to use autocorrelation as an accuracy measurement of simplification seemed to be counterproductive, because for some configurations (e.g. (ε1, ζ_7, η_1)) it did not decrease but increased, at least until exp#70. Moreover, three different products were chosen with different demands (product 14 – demand 35 lots/week, 12 – 5 lots/week, 1 – 2 lots/week) to show that autocorrelation varies for different products depending on the quantity of the product lots in the model.

The forth typical metric is a t-test (Piplani and Puah 2004). First of all, it is necessary to mention that it is recommended to use a t-test if there is a normal distribution. Our proof of the normality using R package normtest (Gavrilov and Pusev 2014) was unsuccessful. Nevertheless, it could still be interesting to see the results of t-tests within this research (Figures 7a and 7b). For the configuration (ε_1, ζ_1, η_1) for both products we see the t-statistics increase and then fall. This is unsurprising because there is a shift of lot CT mean (see Figure 4b). Moreover, the p-value is above 0.2 for product 1 even for exp#83 and we have significant oscillations of the p-value from exp#1 to exp#83. In the end, it can be concluded that t-test is not suitable for the research being carried out in this paper.

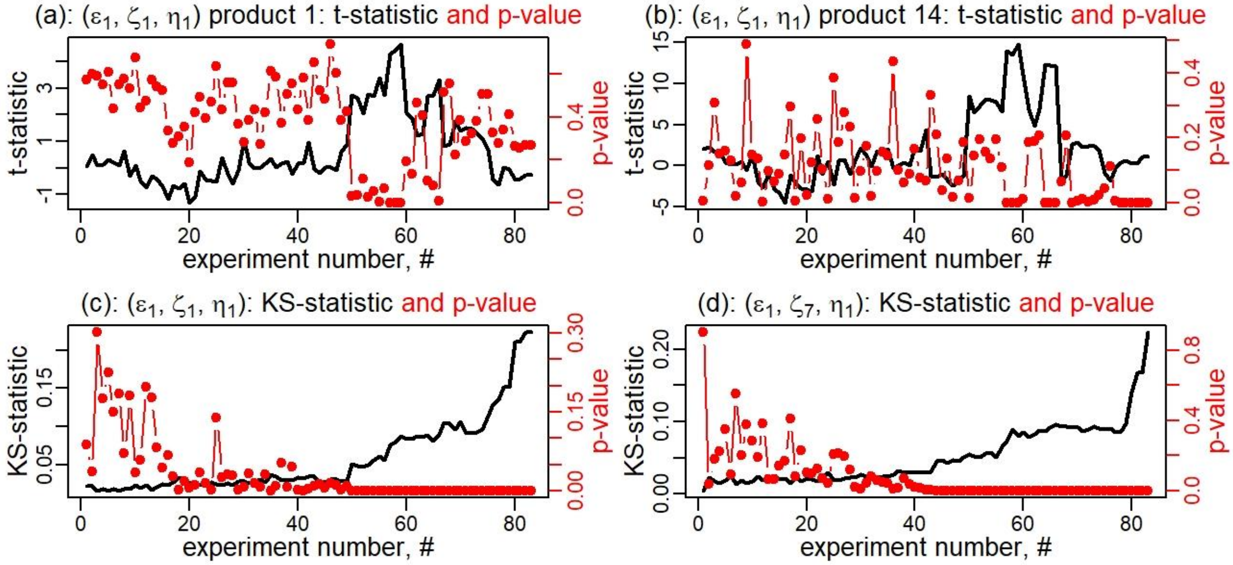

Figure 7: (a) and (b) t-test, (c) and (d) Kolmogorov-Smirnov test: statistic (black) and p-value (red).

4.2 More suitable metrics

Now let us look at the results of a classical KS-test (Figures 7c and 7d). Here it is clear that the KS-statistics also have some fluctuations, but they are much smaller than the t-statistics and, more importantly, the KS-statistics do not grow monotonously but quite strongly. The p-value also has oscillations but it tends to zero quite quickly (compare with Figures 7a and 7b). It is assumed that using metrics based on lot CT distribution is more suitable for the simplification task. First of all, they are not sensitive to the "butterfly effect", which occurs at exp#1 (see Figure 5b), because they consider all lots as a whole (distribution) as opposed to just looking at a single one. Second, the "pathological" (non-monotonous) behavior due to lot CT mean shift is not as expressed (see Figure 4b), because the metrics are not as sensitive to the shift.

In the Section 3.3 we introduced six metrics based on lot CT distribution. Let us compare them using the same configurations as before: (ε_1, ζ_1, η_1) and (ε_1, ζ_7, η_1) (Figures 8 and 9). Figure 9 shows SAD has the least fluctuations in comparison with other metrics. This occurs due to the use of the probability density function (PDF), which is built as a histogram with equidistant bins (Figures 4c and 4d). In this case, small changes inside one bin do not have an influence on the SAD value. On the other hand, we could see that SAD is less sensitive in the area exp#1-70 in comparison with other metrics. The KS test shows moderate fluctuations and is more sensitive to the changes in the area exp#1-70. This happens as a result of the empirical cumulative distribution function (ECDF), which contains data about every single lot. But the KS-statistic is based on a measurement only of the maximum of the difference between the ECDFs, meaning the measurement of the difference is made only at one point. Before the experiments it was assumed that this is not adequate and the preference was to include the whole ECDF. Such metrics include AD, CVM, DTS, and WASS.

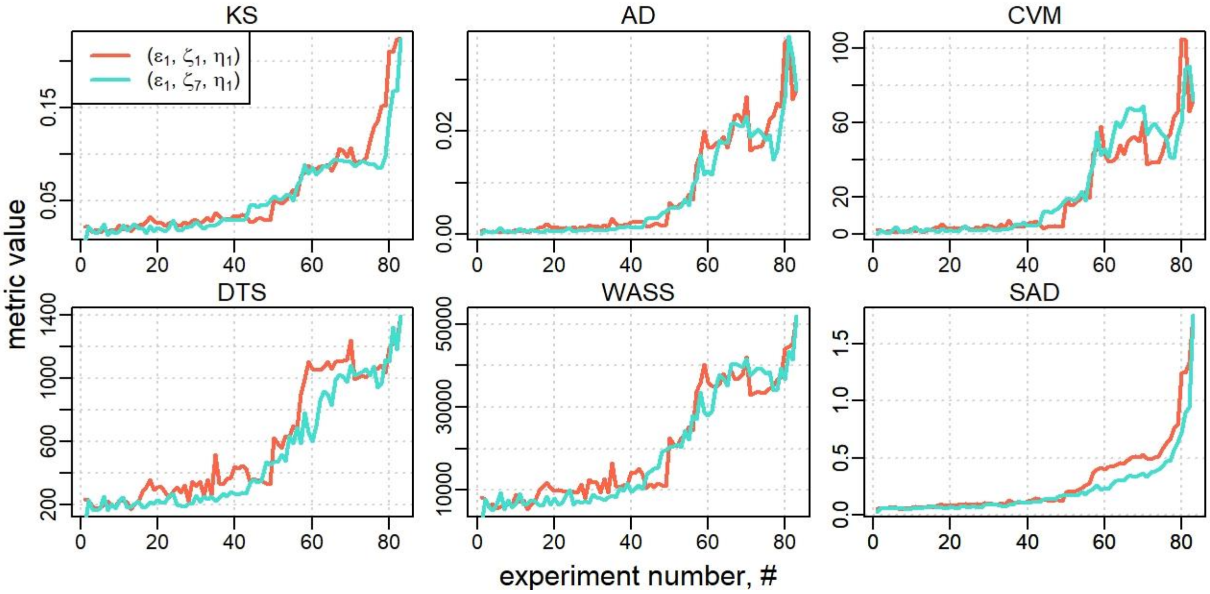

Figure 8: Six metrics in dependence on experiment number.

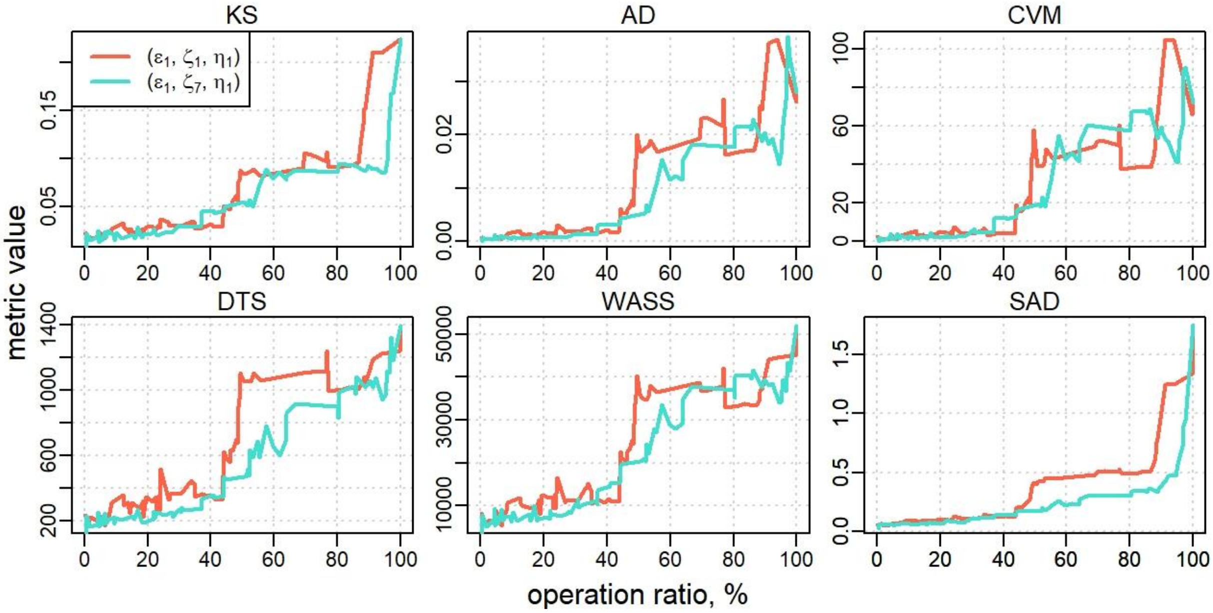

Figure 9: Six metrics in dependence on operation ratio.

It is interesting to see that AD and CVM were less sensitive than KS in the area exp#1-40, and DTS and WASS were somewhat more sensitive than KS in the same area. Moreover, AD and CVM have larger oscillations in the area exp#70-83 than other metrics. These differences between AD/CVM and DTS/WASS are caused by using a factor of 2 in the formulas (Table 1). Moreover, AD/CVM become much more sensitive to the lot CT mean shift as exp# grows. At the end (exp#80-83), even small changes in lot CT mean cause big changes in AD/CVM (compare with Figure 4b). It is assumed that KS and SAD are more reliable metrics for simulation simplification, but future research is needed to prove this statement. Figure 8 represents a picture of the quality. For the calculations, another x-axis was needed – operation ratio (see Figure 9).

As mentioned above, the simulation time curve varies for different configurations (Figure 3b). This means that experiment numbers from one configuration are not equivalent to other configurations in the sense of simplification. We also pointed out that the curve is not monotonous. Therefore we could not use it for the comparison of configurations. On the other hand, it was found that simulation time and operation ratio curves matched sufficiently. Thus we decided for this paper to use the operation ratio as a basis for the comparisons. Figure 9 presents the metrics with operation ratio as the x-axis. Here the difference between two configurations is even more clear than in Figure 8.

5 EXPERIMENTAL RESULTS

To compare various configurations, measurements were made along the axis (metric, operation ratio). Those were then approximated with lines to produce the curves (see Figure 9), which were then numerically integrated to obtain a single value (area under the curve). This value represents a score of the particular configuration in a particular metric. All configurations were ordered according to the score for each metric. To compare two system configurations, ε_2 and ε_{2b}, a single list with those configurations was created. Table 2 shows the results for the first twenty items in each list. For ε_2, a gray background is used. The other colors in the table are only there to easily distinguish different configurations.

Table 2: Ordered configurations.

	FIFO (ε_1)													CR (ε_2) and CRb (ε_{2b})											
#	KS		AD		CVM		DTS		WASS		SAD			KS		AD		CVM		DTS		WASS		SAD	
	ζ	η	ζ	η	ζ	η	ζ	η	ζ	η	ζ	η		ζ	η	ζ	η	ζ	η	ζ	η	ζ	η	ζ	η
1	**4**	1	**8f**	1	**8f**	1	**4f**	1	**4**	1	**8**	1		**8f**	1	**8f**	1	**8f**	1	**8**	1	**8**	1	**8f**	1
2	**4f**	1	**8**	1	**8**	1	**4**	1	**4f**	1	**8f**	1		**8**	1	**8**	1	**8**	1	**8f**	1	**8f**	1	**8**	1
3	**8**	1	**4**	1	**4**	1	**8f**	1	**8f**	1	**4**	1		**4f**	1	**4f**	1	**4f**	1	**7f**	1	**4f**	1	**8**	2
4	**8f**	1	**4f**	1	**4f**	1	**8**	1	**8**	1	**4f**	1		**4**	1	**4**	1	**4**	1	**4f**	1	**4**	1	**4f**	1
5	**4**	2	**5**	1	9	1	**4**	2	**4**	2	**8**	2		**8**	2	**7f**	1	**7f**	1	**4**	1	**8**	2	**8f**	2
6	**4f**	2	**5f**	1	9f	1	**5f**	1	9	1	**8f**	2		**8f**	2	**7**	1	**7f**	1	**8**	2	**8f**	2	**4**	1
7	**8**	2	9	1	**5**	1	**4f**	2	**4f**	2	**4f**	2		**7f**	1	**7f**	1	**7**	1	**8f**	2	**7f**	1	**4f**	2
8	**8f**	2	9f	1	**5f**	1	**5**	1	**8f**	2	**4**	2		**7f**	1	**7**	1	**8f**	2	**7f**	2	**7f**	2	**4**	2
9	**5**	1	**8f**	2	**8f**	2	**8f**	2	**8**	2	**5f**	1		**4f**	2	**8f**	2	**7**	1	**7**	1	**7f**	1	**7f**	1
10	**5f**	1	**8**	2	**8**	2	**8**	2	9f	1	**5**	1		**4**	2	**7f**	2	**8**	2	**7**	2	**7**	2	**7**	1
11	**6f**	1	**4f**	2	**4**	2	9	1	**5f**	1	**6f**	1		**7**	1	**6f**	1	**7f**	2	**7**	1	**7**	1	**7f**	2
12	9	1	**4**	2	**4f**	2	9f	1	**5**	1	**7f**	1		**6f**	1	**8**	2	**6f**	1	**7f**	1	**7**	1	**7f**	1
13	9f	1	**6f**	1	3f	1	3f	1	3f	1	9	1		**5f**	1	**7**	2	**7**	2	2f	2	2f	2	**5f**	1
14	**6**	1	3f	1	3	1	3	1	3	1	**6**	1		**7f**	2	2f	1	2f	2	2f	1	**6f**	1	**6f**	1
15	3f	1	3	1	**6f**	1	**6f**	1	**6f**	1	3f	1		**7**	1	**5f**	1	**6**	1	**5**	1	2f	1	**7f**	2
16	3	1	**6**	1	**6**	1	**5f**	2	**5f**	2	3	1		**5**	1	**5**	1	**5f**	1	**8f**	1	**4f**	2	**8f**	1
17	**6f**	2	**5f**	2	**5f**	2	**5**	2	3f	2	9f	1		**5f**	1	2f	2	2f	1	**6f**	1	**2**	2	**5f**	1
18	**6**	2	**6f**	2	**6f**	2	3f	2	3	2	**7**	1		2f	2	**6**	1	**5f**	1	**5f**	1	**7**	2	**8**	1
19	**7**	1	**7f**	2	**5**	2	**6f**	2	**5**	2	3f	2		**7**	2	**5f**	1	**4f**	2	2	2	**7f**	2	**5**	1
20	**5f**	2	**7f**	1	**7f**	2	9	2	**6f**	2	3	2		**6**	1	**6f**	1	**5**	1	**7f**	2	**4**	2	**7**	2

Table 2 shows that there are different orders for different metrics. Nevertheless, one could see that configurations (ζ_x, η_1) mostly outperform (ζ_x, η_2): operation-based delays are mostly better than machine tool-based. The situation with (ζ_x) and (ζ_{xf}) is not so clear. It seems that those configurations do not differentiate from each other much, meaning it is difficult to evaluate which configuration is superior. For ε_1 and ε_{2b} configurations, (ζ_4, η_1), (ζ_{4f}, η_1), (ζ_8, η_1), and (ζ_{8f}, η_1) produced the best performances. But for ε_2 there were other leaders, namely (ζ_7) and (ζ_{7f}). Based on our previous experience (Stogniy and Scholl 2019), we assumed that ε_{2b} should always outperform ε_2. Table 2 shows that it is not correct. We will continue our research to determine the underlying reasons for this.

CONCLUSIONS

In this paper, a simplification of process flow operations through substitution for constant delays was considered. Numerous experiments with three system configurations were carried out: First In First Out, Critical Ratio, and Critical Ratio with consolidated delays at the beginning. It was found that for the first two system configurations it is better to use queue time or cycle time standard deviation as sieve functions for the substitution. For the third variant, queue length divided by batch size is the best choice.

Different metrics were used to analyze the results of experiments. Based on our analysis we conclude that following metrics could not be effectively used to estimate simplified models: lot cycle time mean absolute error, correlation coefficient, autocorrelation function, and t-test. Metrics based on lot cycle time distributions are more suitable. It is assumed that a classical two sample Kolmogorov-Smirnov test and summarized absolute divergence (based on probability density function) are better than others because they are more stable. Our experiments showed that delays calculated based on average processing and queueing time are not precise enough and could lead to overall lot cycle time shift. This causes oscillation within the metrics. If that problem were to be solved, we assume that metrics based on empirical cumulative distribution functions would show better performance.

In future research we plan to analyze methods for more precise constant delay calculations and to consider other dispatching rules (Early Due Date, Operation Due Date). We will enhance our approach for variable workload scenarios and combine these ideas with the artificial process flow concept, which we presented earlier. Eventually, we plan to implement our approach 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

Alexopoulos, C., and Kelton, W. D. 2017. “A concise history of simulation output analysis”. In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D’Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 115-130. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Dowd C. 2018. twosamples: Fast Permutation Based Two Sample Tests. R package version 1.0.0. https://CRAN.R-project.org/package=twosamples, accessed 20th April 2020.

Gavrilov I., and Pusev R. 2014. normtest: Tests for Normality. R package version 1.1. https://CRAN.R-project.org/package=normtest, accessed 20th April 2020.

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.

Fewster R. 2014. Course notes. STATS 325. Stochastic Processes Department of Statistics University of Auckland. https://www.stat.auckland.ac.nz/~fewster/325/notes/325book.pdf, accessed 20th April 2020

Johnson, R. T., J. W. Fowler and G. T. Mackulak. 2005. “A Discrete Event Simulation Model Simplification Technique”. In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2172–2176. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Law, A. M. 2015. *Simulation Modeling and Analysis*. 5th ed. New York: McGraw-Hill Education

MIMAC Datasets. 1997. http://p2schedgen.fernuni-hagen.de/index.php?id=296, accessed 20th April 2020.

Piplani, R. and S. A. Puah. 2004. “Simplification Strategies for Simulation Models of Semiconductor Facilities”. *Journal of Manufacturing Technology Management* 15(7): 618-625.

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 20th April 2020.

Rose, O. 1999. “Estimation of the Cycle Time Distribution of a Wafer Fab by a Simple Simulation Model”. In *Proceedings of the Semiconductor Manufacturing Operational Modeling and Simulation conference*, San Francisco, CA, USA,133-138.

Roser, C., and Nakano, M. 2015. “A quantitative comparison of bottleneck detection methods in manufacturing systems with particular consideration for shifting bottlenecks”. In *Proceedings of the IFIP International Conference on Advances in Production Management Systems*. 273-281. Springer, Cham.

Stogniy I. and Scholl W. 2019. Using delays for process flow simplification. In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 2372 – 2383. 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.

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. From 1984 to 1995 he worked as a process engineer for ZMD in Dresden. In 1996 he joined Infineon Technologies (formerly SIMEC) and worked in the field of capacity planning. Since 2003 he has been responsible for fab simulation. 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.