Proceedings of the 2000 Winter Simulation Conference
J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

WHY DO SIMPLE WAFER FAB MODELS FAIL IN CERTAIN SCENARIOS?

Oliver Rose

Institute of Computer Science
University of Würzburg
97074 Würzburg, GERMANY

ABSTRACT

Previous work has proved that simple simulation models are sufficient for analyzing the behavior of complex wafer fabs in certain scenarios. In this paper, we give an example where the simple model fails to accurately predict cycle times and WIP levels of the complex model. To determine the reason for this behavior, we analyze the correlation properties of a MIMAC full fab model and the corresponding simple one. It turns out that the simple model is not capable of capturing the correlations in an adequate way because there is lot overtaking (passing) in the simple model while almost no overtaking can be found in the complex counterpart.

1 INTRODUCTION

In some recent papers, simple simulation models are successfully used to explain the behavior of wafer fabs (Rose 1998) or to predict their performance measures such as product cycle times (Rose 1999a). The simple model reproduces only the behavior of the bottleneck workcenter in detail while all other machines are aggregated into delays with adequate distributions. Each lot visits the bottleneck as often as it would do in a real fab. By design, this kind of model mimics the dependence of the fab capacity on the bottleneck properties and the cyclic fashion of production, two of the major constraints of wafer fabrication. We found, however, that as soon as lot release regulation, e.g., CONWIP (CONstant Work In Progress) (Wein 1988, Hopp and Spearman 1991), is implemented the simple model fails in predicting WIP and cycle time of the complex model. This happens despite the fact that the delay distributions of the simple and complex model are almost identical.

We conjecture that correlation between the delays must also have an important impact on the performance of the models. Therefore, we analyze in detail the correlation properties of the MIMAC (Measurement and Improvement of Manufacturing Capacity) full fab model (Fowler and Robinson 1995) and the simple model and reason on the cause of the differences found.

The paper is organized as follows. Section 2 introduces the MIMAC and the simple simulation model. We show the main results from the CONWIP simulation study where the simple model fails. In Section 3, we provide the measures that we use for our correlation analysis. Then, different types of dependences between lots are introduced and examined. Section 4 deals with overtaking of lots as the main reason for the presence of correlation in wafer fab models.

2 SIMULATION MODELS

2.1 MIMAC Models

This section presents the wafer fab simulation models we used to derive our simplified models. As we had no possibility to analyze real world data from a semiconductor wafer fabrication facility we produced such data by using one of the MIMAC model sets and the Factory Explorer simulation tool (Wright Williams & Kelly 1997). These factory-level data sets are freely available at the SEMATECH (SEmiconductor MAnufacturing TECHnology) FTP-Server `ftp://ftp.sematech.org/pub/datasets`.

We chose MIMAC model 6 as reference for our simulation studies. It consists of 228 workstations that perform 104 different processing steps. A total of 9 different products are built in this fab.

The input rate was set to 95% of the approximate maximum that was estimated by Factory Explorer's capacity analysis which is run before the simulation. We did not use lower input rates, because they would decrease the impact and the differences between constant and CONWIP lot release rules. Each tool group in the model uses the FIFO dispatch rule (First In First Out).

Because of modeling restrictions in our simple models we had to make changes to the MIMAC model. First of all we do not model operator availability. Operators are normally used in the production process to load and unload the workstations with lots, to set up the machines and to supervise the production. We assume in our models that there is always an infinite number of operators available.

The second change we implemented focuses on the scrap a machine produces. With a probability of 0.04% each machine makes the lot it currently processes unusable. This happens very rarely and has no considerable impact on the overall fab performance. Therefore, the fab was simplified in such a way that it produces no scrap at all.

As the simple model introduced in the next section only has product sequences, which enter and leave the main bottleneck workstation, we have to remove all products from the MIMAC model, that do not have the bottleneck in their processing sequence. These are the products with numbers 2, 7 and 8. Therefore we have models with six different product types (1, 3, 4, 5, 6, 9). The removal of these 3 products does not change the location of bottleneck.

If we want to study inter-cycle correlation alone we remove all production processes except for product 1. This enables us to switch off inter-product correlation. Of course it is possible to observe inter-cycle correlation in the original fab, but in this case the correlation curve is always a mixture of both types of correlations.

2.2 Simple Models

In this section, we briefly describe a simple model for a wafer fabrication facility, which was introduced in (Rose 1998). This model intends to reduce the required tool-groups in a fab to a minimum. Nevertheless, this model still tries to capture the behavior of the complex fabrication facility.

2.2.1 Basic Simple Model

Figure 1 depicts the bottleneck workstation which is the most important part in this model and in the real fab because it limits the throughput. It is the workstation named 20540_CAN_0.43_MII in the MIMAC fab 6. This tool-group processes each lot separately. Hence, there is no need to model batches in our simple model and we can use lots as the smallest entity in our models.

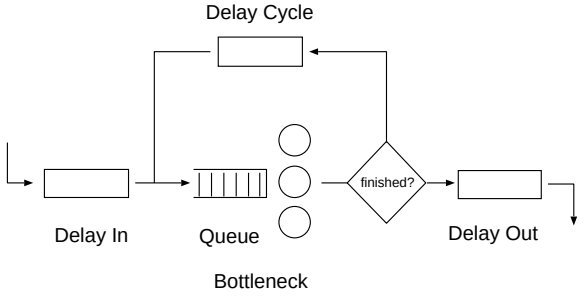

Figure 1: Simple Wafer Fab Model

There are three additional groups. These are the "Delay In" group, the "Delay Cycle" group and the "Delay Out" group. Each group represents the delay a lot has to experience as it passes through the tool-groups including the processing and the waiting time in queues defined by the subsequences in the production process. The "Delay In" group covers the delay a lot needs from entering the fab until it reaches the bottleneck for the first time. The "Delay Cycle" group models the delay a lot needs after leaving the bottleneck and before entering it the next time. It is very likely that the sequence is different for two bottleneck entries. So we need to know how often the bottleneck has been visited by a lot. When a lot completes its final bottleneck processing step it enters the third delay group that is left as soon as the production of this lot is finished.

Next, we take a closer look at the design of the delay groups. As previously mentioned, each group represents a subset of the production process. For each product and group we receive a set of delay times which occurred during the Factory Explorer simulation. Now, it is possible to build histograms from these data sets and use them as a discrete distribution to sample delay values in the simple model. Another way is to estimate the parameters of a given distribution type based on the histogram data. We use Gamma variates because those distribution types cover a wide range of shapes and are a good fit for most of the simulation data. In the simple fab models we apply the three parameter Gamma distribution Gamma(α, β, γ) with shape, scale, and location parameters α, β, and γ, respectively. It has the density

$$f(x;\alpha,\beta,\gamma) = \frac{(x-\gamma)^{\alpha-1}\exp(-\frac{x-\gamma}{\beta})}{\beta^{\alpha}\Gamma(\alpha)}. \qquad (1)$$

The three parameters are determined by a fitting process such as the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm. A detailed explanation and implementation of this method can be found in Press et al. (1992). Figure 2 shows the distribution of the "delay in" group for the first product and the resulting Gamma density function after the fitting process. The distribution is plotted as a histogram where the vertical line shows the distribution's mean.

All simple models were implemented with the ARENA simulation tool (Kelton et al. 1997).

2.2.2 Simple Model with CONWIP Rule

The simple model has to be modified in order to run the factory with a CONWIP rule (Figure 3). In this new model the lots are released in constant time intervals, but if the upper lot limit for a product is reached all lots of this product have to wait in the queue in front of the fab until a lot of this product finishes processing and leaves the system.

If the lot limits are set too high this model behaves as the basic simple model without WIP control . If the limits are set close to or below the average number of lots

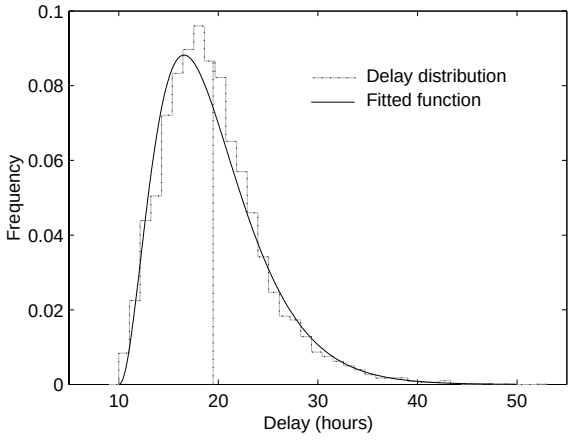

Figure 2: Delay Distribution of Product 1 in the First Delay Group and Gamma Density Function Estimate

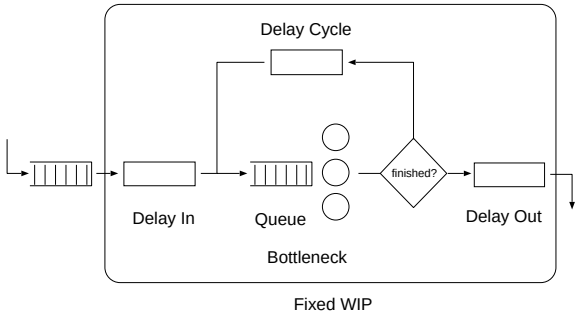

Figure 3: Simple Wafer Fab Model With WIP Control

in the system there is a fixed number of lots in the system. An inventory level can be specified in the Factory Explorer simulations in the same way. Therefore we can compare this release rule for both the complex factory and the simple model.

2.3 Simulation Parameters

For each model the simulated period is 10 years. The first year is considered as the warmup phase and its results are discarded. We required such a long simulation period to obtain enough data for the estimation of all delay distributions of the simple model. For the simulations with CONWIP lot release we defined six models with different WIP levels that are sufficient to analyze the rule's effects on the fab performance. The upper bounds for the inventory are given in Table 1. The model with CONWIP level 25 is similar to a model without CONWIP because the inventory boundaries in models with WIP level 25 represent the maximum value reached in a simulation run without a WIP bound. If the inventory bounds for all six products in the model with CONWIP level 25 are decreased by one, we receive the next lower boundaries of the model with CONWIP level 24.

Table 1: CONWIP Levels

CONWIP level	Levels for product type					
	1	3	4	5	6	9
25	25	13	19	48	34	16
24	24	12	18	47	33	15
23	23	11	17	46	32	14
22	22	10	16	45	31	13
21	21	9	15	44	30	12
20	20	8	14	43	29	11

The boundary values for all other models can be computed in the same way.

2.4 Drawbacks of the Simple Model

In this section, we discuss some limitations and disadvantages of the simple model. Fig. 4 and 5 show the cycle times and inventories of product 1 in the MIMAC and simple models. The solid histograms were built from the MIMAC model data. The dotted ones represent simple model data. The mean values are shown as vertical lines within the histograms. The distributions and means of both models differ. Even though the single delay groups have nearly the same means and distributions due to the fitting of gamma distributions to the simulation data, the MIMAC model has on the average higher cycle times and work in progress. Taking into account the simplicity of the reduced model, however, the accuracy of the above results can be acceptable in certain research contexts.

A drawback arises, however, when using the CONWIP lot release strategy. Fig. 6 and 7 illustrate the differences in the cycle time and inventory of both models for CONWIP level 24. The results for a model with CONWIP level 25 are not depicted because they are almost identical to those of a model with no WIP regulation.

In the MIMAC model the cycle time increases for an inventory level of 24. The inventory is always at the highest

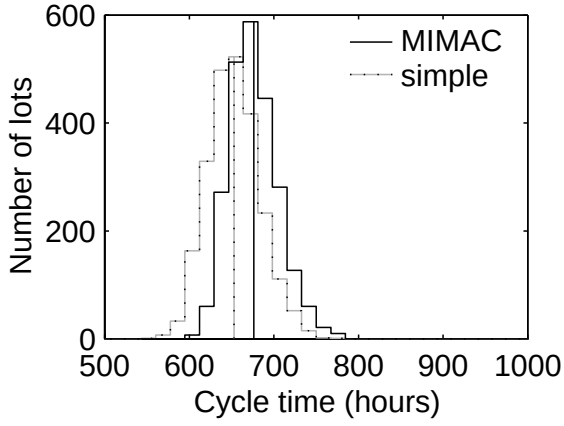

Figure 4: Product 1 Cycle Time Without CONWIP

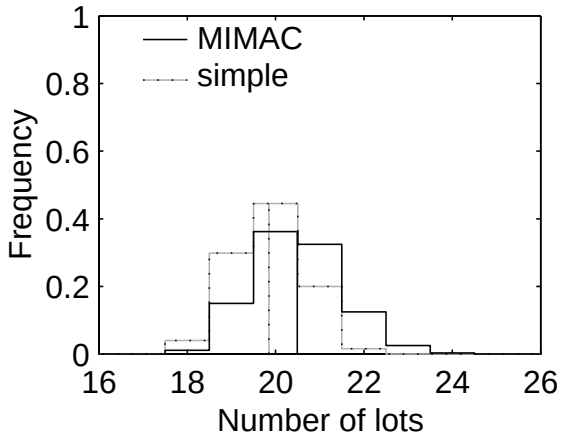

Figure 5: Product 1 WIP Without CONWIP

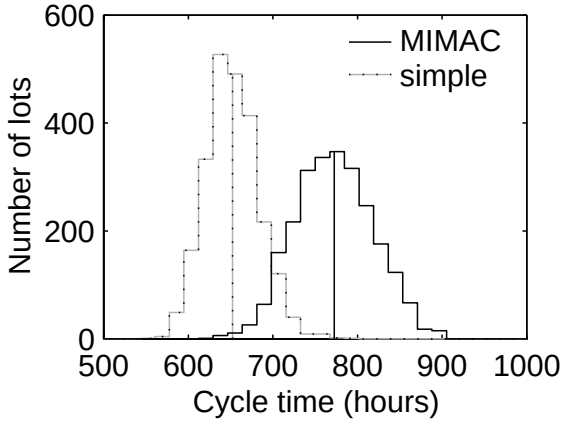

Figure 6: Product 1 Cycle Time With CONWIP Level 24

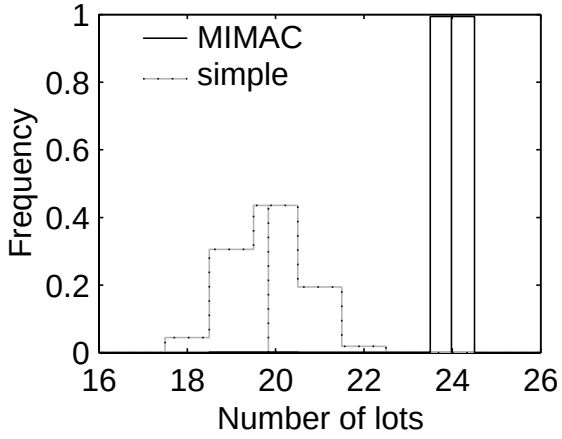

Figure 7: Product 1 WIP With CONWIP Level 24

possible level. The simple model on the other hand shows no change in cycle time distribution and the number of lots is always below the allowed level and it is spread over several WIP levels as in the model with no inventory bound.

As it is not possible to model a CONWIP lot release strategy with the simple model, it is unlikely that the model is able to cope with other lot scheduling strategies as described in Wein (1988) and Rose (1999b).

Consequently, there is a need to further analyze the behavior of a semiconductor facility and to model it in a more detailed way. In particular, the dependences between the products or between the lots of a particular product have a great impact on the factory characteristics and have been neglected by the models so far.

3 CORRELATION IN WAFER FABS

Before we start discussing dependences between delays in the MIMAC and the simple model, we first present the basics for measuring correlation. First, formulae are introduced that we need to calculate correlation between measured data. Then, types of correlation we measured in the wafer fab are presented.

3.1 Correlation Equations

In this section, we define the terms correlation and autocorrelation of stochastic processes or data vectors. These definitions can also be found in most computer simulation or statistics book (Law and Kelton 1991).

Let A be a distribution and E[A] its expectation.

The coefficient of correlation of two distribution functions A_1 and A_2 is expressed as

$$\mathrm{COR}[\mathrm{A}_1, \mathrm{A}_2] = \frac{\mathrm{E}[\mathrm{A}_1 \cdot \mathrm{A}_2] - \mathrm{E}[\mathrm{A}_1]\mathrm{E}[\mathrm{A}_2]}{\sqrt{\mathrm{E}[(\mathrm{A}_1 - \mathrm{m}_1)^2]}\sqrt{\mathrm{E}[(\mathrm{A}_2 - \mathrm{m}_2)^2]}} \quad (2)$$

$$\text{where} \quad \mathrm{m_i} = \mathrm{E}[\mathrm{A_i}], \quad \mathrm{i} \in \{1, 2\}$$

The analysis of our simulation models requires to use data vectors instead of distribution functions A_i in Equation 2. Both data vectors $A_1 = [a_1^1, \cdots, a_n^1]$ and $A_2 = [a_1^2, \cdots, a_n^2]$ must have the same size $|A_1| = |A_2|$. We use notation $A(i)$ to access element a_i of vector A with $1 \leq i \leq |A|$ and $A(i, j)$ to receive or to modify elements a_i up to a_j, where $1 \leq i < j \leq |A|$.

Autocorrelation is a special case of the standard correlation. Instead of using two different data vectors only one is used and the correlation between values within this sequence is expressed by autocorrelation. Autocorrelation is computed for different lags. The lag is the distance between two values for which the correlation value should be gathered.

Let A be a data vector. The autocorrelation of its values is defined by

$$\mathrm{ACOR[A, l]} = \mathrm{COR[A(1, |A| - l), A(l, |A|)]} \qquad (3)$$

$$with\ 0 \leq l \leq |A|$$

3.2 Sliding Windows

For some cases of interest we cannot use the correlation Equation (2) and (3) directly. For example, it is not applicable if we intend to measure the dependences between the delays of product 1 and product 2 while being in cycle 2. It is not possible to use the above coefficient of correlation for these two sequences of delay measurements because the measurement do not happen at the same points in time. In addition, it is unlikely that the number of measurements of each sequence will be the same for a given observation period. Thus, we have to transform these continuous time-dependent data sets into sequences with the same number of values and a discrete index. This problem seems to be uncommon to the statistics community. Therefore, we developed our own approach to transform the raw data sets into a graph that can be used for comparing the correlation properties of the considered models.

In Figure 8, two different time dependent data sequences X and Y are shown. The elements e of each sequence are pairs (e_t, e_v) of a time point e_t when the delay is started and the delay value e_v. In order to calculate a correlation value for the sequences we put a window over both of them. This window starts at a discrete point in time w_t and has a predefined fixed size w_s. The window slides from the beginning of each sequence to its end using W_t and w_s to set a new position. Given a sequence E, let

$$e_t^{min} = \min\{e_t | (e_t, e_v) \in E\}$$

and

$$e_t^{max} = \max\{e_t | (e_t, e_v) \in E\}.$$

Then we build W_t as

$$W_t = [w_t^1, w_t^2, ..., w_t^n], \quad \text{with } w_t^1 = \min\{x_t^{min}, y_t^{min}\} \text{ and } w_t^n \geq \max\{x_t^{max}, y_t^{max}\}.$$

The w_t^i must be subsequent and may not overlap

$$w_t^{i+1} = w_t^i + w_s.$$

Note that for both sequences we have to use the same start time otherwise we would perform a mixture between autocorrelation and correlation computation for two different sequences. Each time the window covers a new range

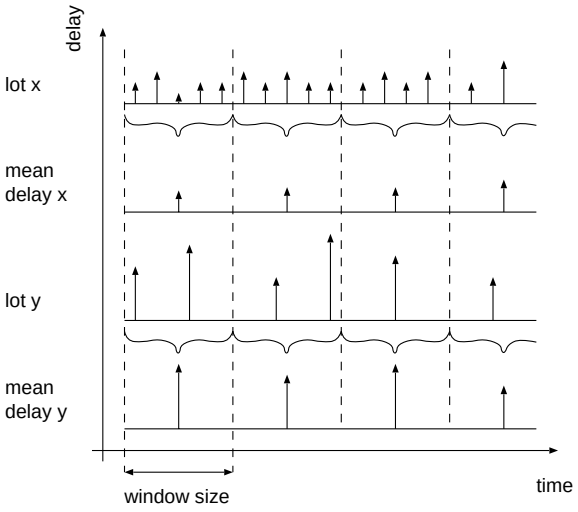

Figure 8: Correlation Computation With Sliding Window

$[w_t^i, w_t^i + w_s)$ all values, which lie therein, are collected from the data sequence.

$$X_w^i = \{x | x = (x_t, x_v) \in X \wedge w_t^i \leq x_t \leq w_t^i + w_s\}$$

As it can be seen from Figure 8 the mean of these values is computed and stored in a new data set $\overline{X}_w$.

$$\overline{X}_w^i = \frac{X_w^i}{|X_w^i|}$$

$$\overline{X}_w = [\overline{X}_w^1, \overline{X}_w^2, ..., \overline{X}_w^n]$$

The same procedure is applied to data sequence Y. By definition both new data sets are of equal size:

$$|\overline{X}_w| = |\overline{Y}_w|.$$

The generated data sets are now used to compute a representation of the correlation that exists between the original data sets by Equation (2)

$$c = \mathrm{COR}[\overline{\mathrm{X}}_\mathrm{w}, \overline{\mathrm{Y}}_\mathrm{w}].$$

Yet, it is not clear which window size w_s should be used to divide the data in equal sized pieces. If the window size is too small there might be windows without any data in it. On the other hand if the window size is too large the averages are computed over too many values and we lose certain time dependent characteristics of the data. In the sequel, we therefore use a graph with a range of window sizes and the resulting correlation values, instead of a single correlation value with a fixed window size. There is no obvious interpretation available for this graph but now we

are able to compare correlation properties of the MIMAC fab with the ones of our simple model.

So far we have only considered the computation of correlation between data vectors. Next, we explain in which cases correlation occurs in wafer fabs and which effects cause this correlation. In most cases we measure the correlation of lot delays between two passes through the bottleneck.

3.3 Correlation in a Single Cycle

In a wafer fab, there is obviously correlation between the delays of consecutive lots of the same type. Figure 9 illustrates this case for a delay cycle and a single wafer type. Wafers with a higher lot number are dependent on those which entered the cycle earlier. The arrows indicate dependences between the lots. If a lot of wafers has to wait a long time in a queue before processing is started, it is very likely that some of its successors have to wait for a longer time, too. As a delay group represents a sequence of tool-groups it is clear that such dependences occur more than once and we only measure and model the aggregation of this lot behavior. This kind of dependence is expressed by autocorrelation, see Equation (3).

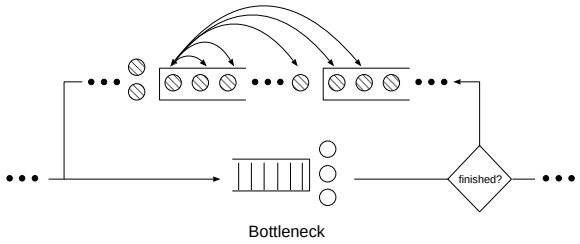

Figure 9: Correlation in a Single Cycle

Now, we compare the autocorrelation properties of both the MIMAC and the simple model. Figure 10 presents only the first delay cycle of product 1 because all other cycles and products behave in a similar way. From the two curves we conclude that there are no dependences between lots in the same cycle in our simple model. On the contrary, the MIMAC model shows significant autocorrelation values for the first two non-zero lags.

3.4 Correlation Between Cycles

In Figure 11 wafer lots in two different cycles are illustrated. Both sequences share a common tool-group in this sketch. In a real fab this could be several workstations. Table 2 illustrates the common tool-groups in the cycles 2 and 3 of product 1. The third column indicates if two tool-groups match (m), if a space has to be inserted (i) or if there is a mismatch. The sequence of all matching tool-groups is the longest common subsequence (LCSS) of both original sequences. The LCSS algorithm is a special case of

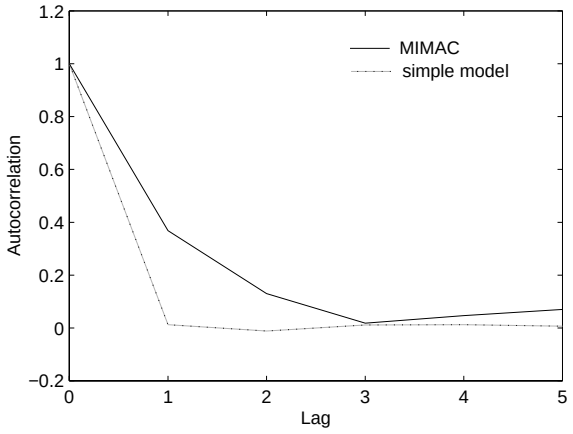

Figure 10: Autocorrelation of Product 1 in the First Delay Cycle

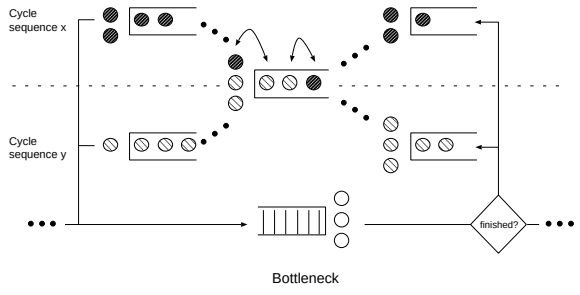

Figure 11: Inter-Cycle, Inter-Product Correlation

the global alignment problem and is described in Gusfield (1997). The number of matches is 13, of mismatches 4, and of gaps 10. Obviously, the match percentage is 48 percent for both sequences. We see, that the dependences are caused by the different types of lots as they are intermixed in the queues and in the workstations. We call these dependences between lots of the same product type, but in different cycles *inter-cycle correlation*.

A good way to illustrate inter-cycle correlation is to use a fab model with only a single product (Figure 12). Once again, the simple model cannot reproduce the correlation values for longer window sizes in the MIMAC model. The correlation values below 30 should not be considered as too important because the mean interarrival time for cycle 2 is 33.07 and for the third cycle is 33.07. The standard deviation is in both cases 10.81. Therefore we conclude, that only a few lot delay events occur in windows smaller than 30.

In summary, the simple model shows no inter-cycle correlation while the MIMAC model exhibits strong dependences among lots of different cycles.

Table 2: Longest Common Subsequence of the First Product

Proc. seq. cycle 2	Proc. seq. cycle 3	
20540_CAN_0.43_MII	20540_CAN_0.43_MII	m
10121_DNS-PUD	10121_DNS-PUD	m
15122_LTS_1	15122_LTS_1	m
	15627_HIT_S6000	i
	13121_LAM_490B_1+2	i
	15121_LTS_3	i
	15123_LTS_2	i
16221_IMP-MC_1+2	16221_IMP-MC_1+2	m
12553_POSI_GP	12553_POSI_GP	m
12531_SH	12531_SH	m
15131_LZZZZ	15131_LZZZZ	m
	15627_HIT_S6000	i
12021_AUTO-CL_undot	12021_AUTO-CL_undot	m
11029_ASM_C1_D1	11027_ASM_B3_B4_D4	
15123_LTS_2	15123_LTS_2	m
12226_NF-2	12226_NF-2	m
12021_AUTO-CL_undot	12240_NIT-ETCH	
11026_ASM_B2	15131_LZZZZ	
15123_LTS_2	15123_LTS_2	m
11125_ASM_E1_E2_H4	12022_AUTO-CL_dot	
15123_LTS_2		i
11127_ASM_E3_G2_H3		i
15123_LTS_2		i
15421_SURF_2		i
10123_DNS-3	10123_DNS-3	m
10123_DNS-3	10123_DNS-3	m

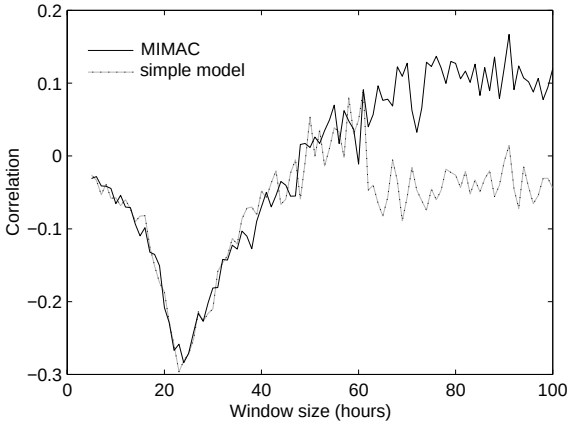

Figure 12: Correlation Between Cycle 2 and 3 in Single Product Models

3.5 Correlation Between Products

The term inter-cycle correlation can be generalized to several products. Both types of wafers in Figure 11 can be considered as different products as well. As several product processes have to share the same sets of tool-groups, correlation between products can be expected. Basically, this *inter-product correlation* is a generalization of inter-cycle correlation. Table 3 gives an example which dependences can exist between two different wafer products. In this case

Table 3: Longest Common Subsequence of the First Product in the Second Cycle and the Third Product in the Second Cycle

Proc. seq. of product 3	Proc. seq. of product 1	
20540_CAN_0.43_MII	20540_CAN_0.43_MII	m
10121_DNS-PUD	10121_DNS-PUD	m
15122_LTS_1	15122_LTS_1	m
	15627_HIT_S6000	i
	13121_LAM_490B_1+2	i
	15121_LTS_3	i
	15123_LTS_2	i
16221_IMP-MC_1+2	16221_IMP-MC_1+2	m
12553_POSI_GP	12553_POSI_GP	m
12531_SH	12531_SH	m
15131_LZZZZ	15131_LZZZZ	m
	15627_HIT_S6000	i
12021_AUTO-CL_undot	12021_AUTO-CL_undot	m
11029_ASM_C1_D1	11027_ASM_B3_B4_D4	
15123_LTS_2	15123_LTS_2	m
12226_NF-2	12226_NF-2	m
12021_AUTO-CL_undot	12240_NIT-ETCH	
11025_ASM_B1_H2	15131_LZZZZ	
15123_LTS_2	15123_LTS_2	m
11125_ASM_E1_E2_H4	12022_AUTO-CL_dot	
15123_LTS_2		i
11127_ASM_E3_G2_H3		i
15123_LTS_2		i
10123_DNS-3	10123_DNS-3	m
10123_DNS-3	10123_DNS-3	m

there are 13 matches, 4 mismatches, and 9 gaps, which gives us a match percentage of 50 percent.

To further emphasize this point, Figure 13 shows the correlation curves between lots of product type 1 and 3, both in the second delay cycle. Obviously, the simple model does not show any inter-product dependences.

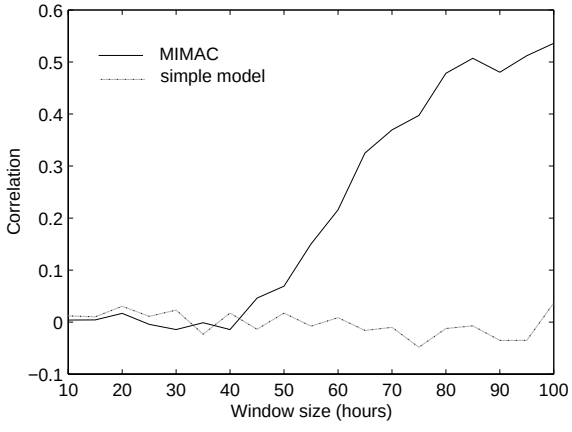

Figure 13: Correlation Between Lots of Product 1 and Product 3 Both in the Second Cycle

Unfortunately, we cannot separate inter-product correlation from inter-cycle correlation. In order to model only inter-product correlation a factory with sequences which have only a single cycle would be required.

4 OVERTAKING

In the previous section, we discovered that the measurements from the MIMAC fab show considerably more dependences of various kinds than the simple model although all delay distributions were approximately the same in both cases. Therefore, there must be at least one feature that is fundamentally different in both models. We came to the conclusion that most of the correlation effects must have been caused by the non-overtaking property of the lots of the MIMAC fab. In contrast, there was no overtaking regulation in the simple model. Successive lots of a certain type receive random delay cycle values from a given Gamma distribution.

Therefore, it is unlikely that a lot always finishes the delay before the lots with a higher lot number. But in the real fab all lots have to proceed through the same tool-groups and overtaking in a queue is not possible due to the FIFO dispatching rule.

4.1 Definitions

In this section, we introduce the terms, *finished delay*, *rest delay*, *relative finished delay*, and *overtaking*, that are used to describe overtaking behavior.

Let x be a delay given as a tuple (x_t, x_v), where x_t is the point in time the delay starts and x_v is the duration of the delay. The *finished delay* of x at point in time t is defined as

$$x_f(t) = \min\{t - x_t, x_v\}, \quad t \geq x_t.$$

The *rest delay* of x is given by

$$x_r(t) = x_v - x_f(t).$$

or

$$x_r(t) = \max\{x_v - (t - x_t), 0\}.$$

The *relative finished delay* x_{rf} is defined as:

$$x_{rf}(t) = \frac{x_f(t)}{x_v}$$

The relative finished delay is a measure of the percentage of a delay, that has already been worked off.

Let x and y be delays. Delay x *is taken over by* delay y if the rest delay of x at the event time y_t is greater than the total delay of y, which is

$$x_r(y_t) > y_v.$$

Figure 14 illustrates the term finished delay and rest delay graphically at the point in time y_t. Lot x is overtaken by lot z but not by lot y.

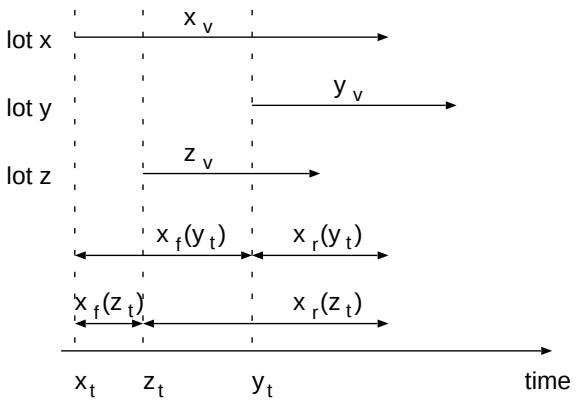

Figure 14: An Overtaking and a Non-Overtaking Case

We see that there are two factors which influence overtaking. The first factor is the length of a delay. The longer a delay is, the easier it is for other lots to overtake, because it is more likely for their delays to fit into the long delay. The second factor is the percentage amount of the finished delay. If the percentage is high, there is only a short rest delay. Therefore it is unlikely that other lots overtake this one. On the other hand, if the percentage is low, the delay just started and the lot needs a longer time to finish. Then future lots have a higher chance to be delayed less than the rest delay of the lot currently waiting and take it over. The relative finished delay allows us to combine both factors in a single measure. The finished delay and the rest delay cannot be used for this purpose as these measures do not take into account the length of the delay.

The *overtaking percentage* of two lot types x, y is defined as

$$OV(x, y) = \frac{|D^y_{\mathrm{OV}}|}{|D^y_{\mathrm{OV}}| + |D^y_{\mathrm{NOV}}|},$$

where $|D^y_{\mathrm{OV}}|$ is the number of overtaking lots and $|D^y_{\mathrm{NOV}}|$ the number of non-overtaking ones. If $p = OV(x, y)$, p percent of the lots of type y overtake the lots of type x.

The overtaking percentage facilitates the comparison of the overtaking behavior of lot types in different simulation models. The overtaking percentage is dependent on shared processing tool-groups and on the delay distributions of both lot types. For instance, if the distribution of lot type x has on the average smaller delay values compared to the values of lot type y, $OV(x, y)$ has a value below 50 percent because most of the time lots of y have a greater delay value than the rest delay of the current x delay. On the other hand, $OV(y, x)$ is above 50 percent as x delays occur more frequently and with shorter delays.

4.2 Overtaking Behavior in the Simulation Models

Now we compare the simple models and the MIMAC model with reference to lot overtaking. The following plots illustrate the entry times of several lots into the bottleneck (Fig. 15 and 16). Each row shows all entry events for a single lot. Again, all models make use of all six products, but only the first one is shown. Each vertical line represents an entry event of a lot. They have been plotted alternately with different line types to facilitate a comparison of neighboring lots. The distance between entry events of the same lot number are the sum of the bottleneck processing time and the delay in a single cycle.

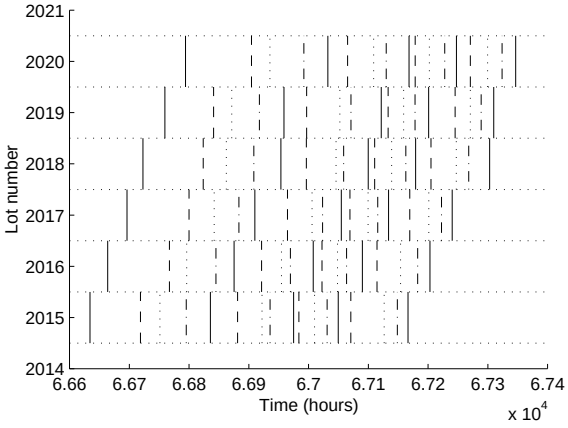

Figure 15: Lots Entering the Bottleneck in the MIMAC Model

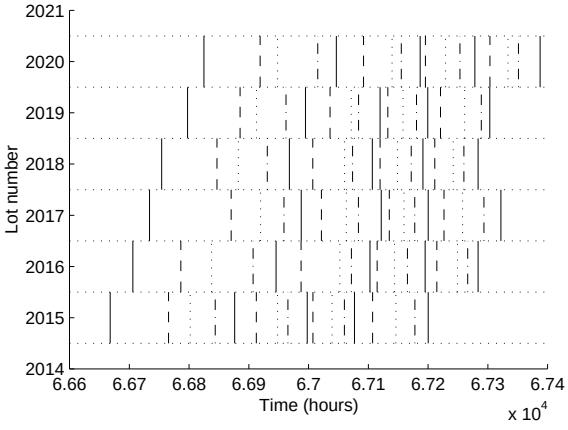

Figure 16: Lots Entering the Bottleneck in the Simple Model

If we look at all vertical lines at the i-th position in the rows, we can compare the overtaking behavior of lots of the same type. Obviously, lots with a higher lot number enter the bottleneck after a given cycle i always after all lots with lower lot numbers have entered the bottleneck workstation in that cycle i. We conclude, that there is nearly no overtaking for lots of the same product type and in the same cycle in the MIMAC model (Figure 15).

If we compare this observation with results from models based on the simple model we observe considerable overtaking in those models. For instance, if we consider lot number 2017 and 2018 on the second entry into the bottleneck (Figure 16). In this case lot 2018 is overtaking lot 2017, because it enters the bottleneck earlier than previous lot. This holds for all following bottleneck entries in this case, too.

As these plots show only a small snapshot, we have counted the number of overtakings and divided it by the sum of overtakings and non-overtakings. The method to generate these overtaking percentages $OV(x, y)$ from the MIMAC data sets is given below. The following steps have to be performed for all possible pairs of lot types x, y.

1. Reset the counters $c_{\mathrm{OV}} = 0$ and $c_{\mathrm{NOV}} = 0$.
2. Let D^i be a vector of delay values and T^i the corresponding time events of lot type $i \in \{x, y\}$.
3. For all elements $t_i^x \in T^x$
   (a) Generate the set of indices
   $$J = \{j | t_i^x \leq t_j^y < t_{i+1}^x\}.$$
   (b) Create the set
   $$D_J^y = \{d_j^y | j \in J\}$$
   (c) Create the sets of overtaking and non-overtaking lots
   $$D_{\mathrm{OV}}^y = \{d_j | d_j \in D_J^y \wedge d_r^x(t_j^y) > d_j\}$$
   and
   $$D_{\mathrm{NOV}}^y = \{d_j | d_j \in D_J^y \wedge d_r^x(t_j^y) \leq d_j\},$$
   where the rest delay is computed by $d_r^x = t_j^y - t_i^x$.
   (d) Update the counters c_{OV} and c_{NOV}:
   $$c_{\mathrm{OV}} = c_{\mathrm{OV}} + |D_{\mathrm{OV}}^y|$$
   $$c_{\mathrm{NOV}} = c_{\mathrm{NOV}} + |D_{\mathrm{NOV}}^y|$$
4. Calculate the overtaking percentage by
   $$OV(x, y) = \frac{c_{\mathrm{OV}}}{c_{\mathrm{OV}} + c_{\mathrm{NOV}}}$$

Applying this procedure we receive the percentage of overtaking lots that occur during a simulation run for models with all six products. Table 4 shows the overtaking

percentages of lots first product. Each column represents a different cycle. The other products behave in a similar way.

Table 4: Overtaking Percentages of Lots of Same Type

	Cycle									
Model	0	1	2	3	4	5	6	7	8	9
MIMAC	0	0	0	0	0	1	1	0	1	0
simple	0	4	5	6	4	6	8	3	5	2

	Cycle							
Model	10	11	12	13	14	15	16	17
MIMAC	1	0	1	0	1	0	1	1
simple	3	4	4	7	8	6	7	12

Since all lots of a certain type have to advance through the same tool-groups and wait in the same queues, it is obvious that they usually cannot overtake each other. This is due to the fact that a FIFO dispatching rule is used in the simulations. In the rare case when a lot is overtaken by another one, it is likely that both lots entered a tool-group at the same time consisting of more than a single workstation. After the processing is finished, they are unloaded from the workstations in the reverse order and an overtaking happens. The same effect occurs if a workstation operates on batches of lots. Both cases are very unlikely and therefore we conclude that there is almost no overtaking in the MIMAC model. In contrast, the simple model shows a considerable amount of overtaking in almost all cycles.

5 CONCLUSIONS

In this paper, we showed that simple simulation models for wafer fabs may lead to questionable performance predictions in certain scenarios. This is caused by the absence of lot dependences that can be found in full fab simulation models like the MIMAC data sets. The main reason for this misbehavior is the fact that overtakings of lots happen due to replacing sequences of work centers by delays. In contrast, overtaking happens only sporadically in a real fab.

We conclude, that in spite of the problems discussed in this paper simple models are still useful but they have to be improved for certain problem scenarios. The problem is to avoid lot overtaking in the simple model without increasing the model complexity too much. This could be done by replacing the delays either by a series of pseudo work stations or by designing a way of sampling the delays such that overtaking of lots is avoided.

ACKNOWLEDGMENTS

The author would like to thank Christian Schloter for fruitful discussions and his valuable programming efforts.

REFERENCES

Fowler, J. and J. Robinson. 1995. Measurement and improvement of manufacturing capacities (MIMAC): Final report. Technical Report 95062861A-TR, SEMATECH, Austin, TX.

Gusfield, D. 1997. *Algorithms on Strings, Trees, and Sequences: computer science and computational biology*. Cambridge University Press, Cambridge, Massachusetts, first edition.

Hopp, W. J. and M. L. Spearman. 1991. Throughput of a constant work in process manufacturing line subject to failures. *International Journal of Production Research*, 29(3):635–655.

Kelton, W. D., R. P. Sadowski, and D. A. Sadowski. 1997. *Simulation with Arena*. McGraw–Hill, New York.

Law, A. M. and W. D.Kelton. 1991. *Simulation Modeling & Analysis*. McGraw–Hill, New York, 2nd edition.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 1992. *Numerical Recipes in C: the art of scientific computing*. Cambridge University Press, Cambridge, Massachusetts, second edition.

Rose, O. 1998. WIP evolution of a semiconductor factory after a bottleneck workcenter breakdown. In *Proceedings of the Winter Simulation Conference '98*.

Rose, O. 1999a. Estimation of the cycle time distribution of a wafer fab by a simple simulation model. In *Proceedings of the SMOMS '99 (1999 WMC)*.

Rose, O. 1999b. CONLOAD - A new lot release rule for semiconductor wafer fabs. In *Proceedings of the Winter Simulation Conference '99*.

Wein, L. M. 1988. Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing*, 1(3):115–130.

Wright Williams & Kelly. 1997. *Factory Explorer 2.3 User Manual*.

AUTHOR BIOGRAPHY

OLIVER ROSE is an assistant professor in the Department of Computer Science at the University of Würzburg, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from the same university. He has a strong background in the modeling and performance evaluation of high-speed communication networks. Currently, his research focuses on the analysis of semiconductor and car manufacturing facilities. He is a member of IEEE, ASIM, and SCS. His email and web addresses are, respectively, <rose@informatik.uni-wuerzburg.de> and <www-info3.informatik.uni-wuerzburg.de/~rose>.