Cycle Time Prediction for Semiconductor Manufacturing via Simulation on Demand

Bruce Ankenman
Barry L. Nelson
Mustafa Tongarlak
Northwestern University

John Fowler
Gerald Mackulak
Detlef Pabst
Arizona State University

Feng Yang
West Virginia University

December 21, 2007

Abstract

Traditionally, competition between semiconductor manufacturers has primarily focused on product design and cost. Recently, speed of delivery has also become an important differentiator among these firms which has led to manufacturing cycle time becoming a critical performance measure. This paper presents a methodology that performs a limited set of simulation runs for a complex wafer fabrication system and then uses the results to develop metamodels that predict mean steady-state cycle time as a function of product mix and throughput. These predictions can be made on demand, i.e., without performing any additional simulation runs, for product mixes and throughput levels not previously simulated. The goal is to support medium and long-range planning by providing results with the fidelity of a detailed simulation model but with the speed of a queueing approximation or simple capacity model.

1 Introduction

Planning for semiconductor manufacturing, either at the factory or enterprise level, requires answering a large number of what-if questions involving different scenarios for product mix, production targets and capital expansion. A key performance measure for evaluating these scenarios is the time. Accurate cycle-time estimation results in a more stable production environment (Chung and Lai 2006). Shorter cycle times may also lead to in the production of a higher quality product and improved responsiveness to customer needs (Hopp and Spearman 2000). Leachman (2002) indicated that "there is considerable evidence that yields are inversely related to manufacturing cycle times."

The importance of cycle time to the semiconductor manufacturing industry is reinforced within the International Roadmap for Semiconductors 2006 Update (Semiconductor Industry Association, 2006); it states that the improvement of cycle-time targets must be met to prevent slowing of the industry's growth. Cycle-time reduction is listed in the road map as a difficult challenge for both the near term and the long term. Nemoto et al. (2000) demonstrate that significant financial benefits come from cycle-time reduction in the ramp-up phases of semiconductor manufacturing. Boebel and Ruelle (1996) and Pfund et al. (2006) also indicate that cycle time has become a key performance metric for semiconductor manufacturers.

The importance given to cycle time as a performance metric in the semiconductor manufacturing industry motivates the need for a quick and accurate way to estimate it. A typical factory must constantly review how proposed changes to product mix, start rates, and process routings will impact the cycle time of both in-process and planned future production. Planners in some semiconductor manufacturing facilities use a Cycle Time-THroughput (CT-TH) curve as a way to quantify the relationship between cycle time and capacity (Fowler et al. 1997). A CT-TH curve (e.g., Figure 1) plots the predicted long-run average cycle time versus throughput (or start rate).

The following example illustrates the importance of accurate and timely cycle-time estimates. Although the dollar values are not specific to any manufacturer they are close enough to reality to illustrate the point.

1.1 Impact of Inaccurate CT Estimates

Suppose that a semiconductor manufacturer can sell their product for $15,000 per wafer, and that their customer base is satisfied with a six week average delivery time. First consider the consequences of *overestimating* the true mean CT-TH curve, as illustrated in Figure 1. Since the manufacturer needs to meet the 6 week average cycle-time constraint, their estimated curve (see point A) will cause them to initiate a start rate of 24,500 wafer starts per month (WSPM). In actual operation, they will achieve a 3 week mean cycle time (see point C). In other words, they will under load the factory. They could have started 25,000 lots (see point B) and still achieved the desired average cycle time. Even though this error is relatively small, 24,500 instead of 25,000 (an error of 2%), the financial loss can be quite large. Consider that it will take at least 1.5 months for the manufacturer to realize that they have underestimated their capabilities. During that time they will have delivered 24,500 WSPM × 1.5 months = 36,750 wafers, generating revenue of $551.25 million. However, had they known their true CT-TH curve, they could have delivered 37,500 wafers generating $562.5 million in revenue. This represents approximately $11.25 million in lost revenue over a single 1.5 month production period. If the error in start rate is not corrected, the annual revenue loss will be much greater.

Next consider what happens if a firm *underestimates* the mean CT-TH curve. If the manufacturer plans on a mean cycle time of 6 weeks, the estimated curve in Figure 2 implies that they can launch 25,000 WSPM and achieve that average cycle time (point A on Figure 2). However, when they implement that start rate the true CT-TH curve indicates

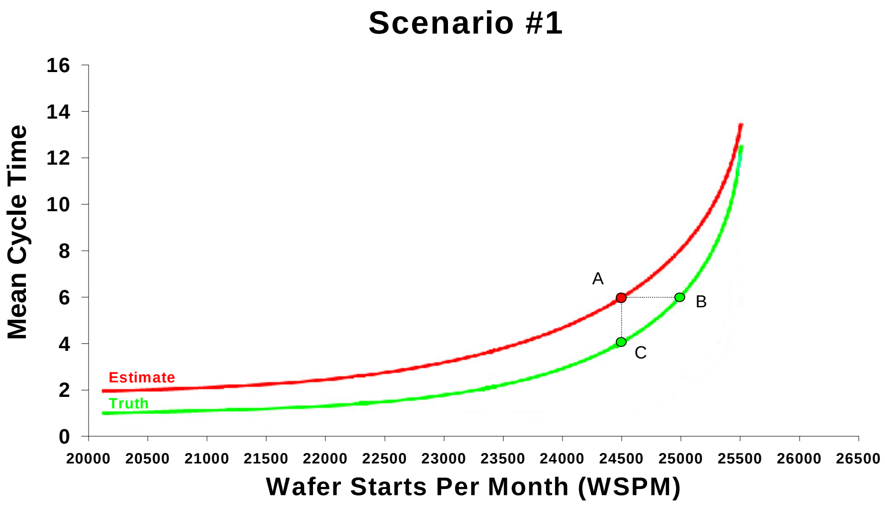

Figure 1: Illustration of overestimating the CT-TH curve (cycle time in weeks).

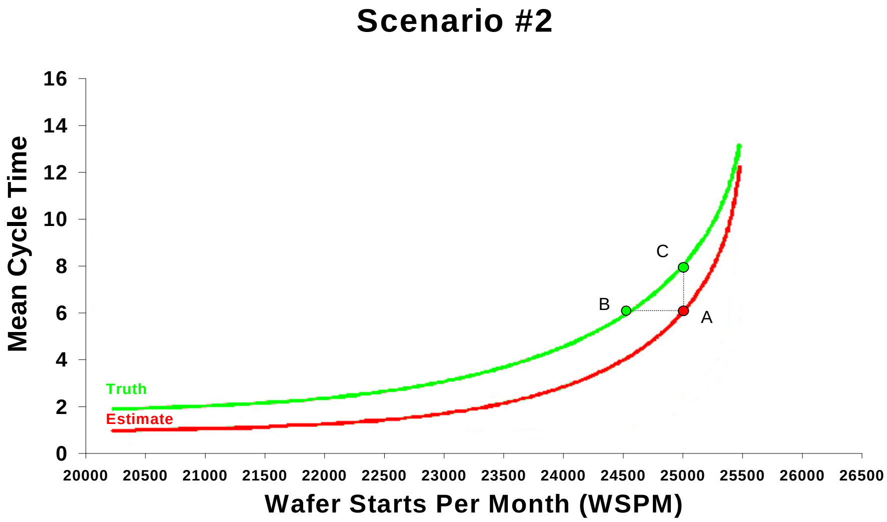

Figure 2: Illustration of underestimating the CT-TH curve (cycle time in weeks).

an average delivery time of 7 weeks (point C). Had they known the true CT-TH curve, then they would have operated at 24,500 WSPM (point B), rather than 25,000.

The economic impact of this error depends on the distribution of cycle time. Assume the distribution of cycle time is symmetric about the mean and the standard deviation is approximately $\sigma = 1/2$ of a week. Since the company is actually operating at point C (7 week average delivery time) their assumption of a 6 week average delivery time (point A) is 2σ from the true value. This implies that an additional approximately 47% of their orders will now be classified as late and may incur penalties. The increase in throughput results in $11.25 million in additional revenue per production cycle; however, penalties and loss of goodwill from missed delivery dates could be devastating. In fact, a penalty rate greater than 4.25% of revenue would completely negate the increased revenue from the higher start rate. An inaccurate mean cycle time estimate in semiconductor manufacturing planning can clearly lead to substantial revenue loss.

1.2 Generation of Cycle-Time Estimates Using Simulation

Manufacture of the more complicated semiconductor devices can routinely consist of 300–500 different processing steps. Each step will be performed by a specific tool group. A factory can have upwards of 135 of these tool groups and a total tool set typically consists of more than 400 machines. A factory may have five or more distinct technologies (or product families) simultaneously competing for these tool groups. Simulation models of such large and complicated systems are by necessity large and complicated themselves.

As discussed above, the impact of using an inaccurate cycle-time estimate for product planning can be financially devastating. The problem is that decisions regarding what to set for product mix (PM) and start rates need to be evaluated within a small window of opportunity. The complexity of a semiconductor factory makes simulation the most likely tool to be used for predicting cycle time, but the very nature of simulation analysis techniques and execution run times make it an insufficient tool if decisions are required within this window of opportunity. Most semiconductor manufacturers have spent significant resources on the creation of extremely detailed simulation models of their facilities only to discover that the run time and analysis issues make them inappropriate for interactive planning exercises.

If it were possible to generate a robust queueing model that reasonably represented the manufacturing system, it would generally be a superior approach to simulation. The queueing model could be evaluated quickly under various start-rate product-mix combinations, producing results as needed. However, semiconductor manufacturing is too complex to be accurately represented by a tractable queueing model. Instead, we desire a way to produce simulation-quality results as quickly as a queueing model while maintaining the level of detail and accuracy needed in semiconductor planning; we call this *simulation on demand.*

1.3 Simulation on Demand

The difficulties of modeling semiconductor wafer fabs with fast and accurate simulation models have motivated researchers to pursue methods for increasing the efficiency of model building and the speed of model execution (e.g., Fowler et al. 2001, Fowler et al. 2005, Mackulak et al. 2005, Park et al. 2002). Incorporating simulation into a planning role requires significant reductions in both run time and experimentation effort. Unfortunately, most of these studies have not provided the order of magnitude improvement to what remains the most serious impediment to using simulation for medium to long-term planning in manufacturing: *it still takes too long to run a full sized factory simulation model.*

The “simulation on demand” concept addresses this fundamental weakness of simulation. Simulation on demand focuses on the *efficiency of obtaining useful simulation results, rather than on the efficiency of the simulation itself.* The premise is to exploit the availability of large quantities of idle (perhaps networked) computer resources by exercising a simulation model in advance of the need to make decisions based on it. A sequential procedure guides the simulation through a series of carefully selected design points. We call the results at these design points a *model structure* (MS), and it is input to a *query engine* (QE) that allows the decision maker to investigate options and trade offs on demand. The QE provides answers to questions such as the following: What is the weighted cycle time of the factory at a particular throughput and product mix? What are the feasible values of throughput and product mix such that average cycle-time constraints are met? What is the impact on the cycle times of other products if we increase the throughput of product i to meet increased demand? And what product mix maximizes revenue while keeping average cycle times below required limits when there is more demand than capacity? Together, the MS and QE constitute a *complete response-surface map* (cRSM) of the CT-TH-PM space (Fowler, et al. 2007). The cRSM provides the fidelity of a detailed simulation with the convenience of a simple capacity model.

In this paper we present the methodology that supports the construction of a cRSM for semiconductor manufacturing. The specific implementation details and large-scale evaluation of the software produced for the Semiconductor Research Corporation will be detailed elsewhere. We do, however, provide a numerical example that illustrates the methodology.

2 Background and Approach

The CT-TH curve has long been used for exploring the relationship between start rate and cycle time for a single product or for a given product mix. Leonovich (1994) proposed optimizing WIP, cycle time, and output rate by defining the relationship between WIP and output rate, and then selecting the output rate that corresponded to some WIP target. Spence and Welter (1988) used a similar cycle time versus throughput trade-off curve to define the operational capacity of a given factory. Martin (1994) imposed a cycle-time requirement on a manufacturing line to improve capacity planning. Aurand and Miller

(1997) formalized the use of the CT-TH curve (which they called the *operating curve)* as a way to measure and benchmark overall fab performance. In addition, Schoemig (1999) investigated the impact of variability on the CT-TH (operating) curve.

There have been two main streams of research: The first formalized the use of the "extended X-factor contribution" for capacity planning. The extended X-factor contribution identifies system capacity constraints based on machine group utilization and raw processing time. The seminal research in this area includes papers from Martin (1998, 1999, 2000), Ozawa et al. (1999), Kishimoto and Ozawa (2000), Occhino (2000), and Kishimoto et al. (2001). Recently, Delp et al. (2005, 2006) augmented this approach with an increased emphasis on the role of machine availability and on variability. The second stream of research deals with the efficient generation of the single product (or single product mix) CT-TH curve. Papers in this area include Fowler et al. (2001), Park et al. (2002), Mackulak et al. (2005), and Yang et al. (2007ab).

Our goal is to extend the previous research in the second stream to predict long-run (steady-state) properties of product cycle time—on demand—as a function of product release rates, allowing us to vary product mix as well as throughput. In this paper we focus on mean cycle time, but the tools we develop can also be used to predict higher moments of CT (e.g., standard deviation), and even percentiles of CT using moment-based approximations (Bekki et al. 2007, Yang et al. 2007b). We assume the availability of a detailed fab simulation model and also a corresponding capacity model that can predict the utilization of all major workstations for any given set of product release rates. The simulation model will be used to estimate mean CT while the capacity model will identify the bottleneck station or stations. Determining when the bottleneck switches from one station to another turns out to be critical to our methodology. We also assume that the equipment, personnel and control policies (such as scheduling and dispatching) remain the same within a given scenario regardless of product release rates. Finally, we assume that a "greedy" batch policy (Fowler et al. 1992) is employed in the factory. This last assumption is necessary so that the CT is monotonically increasing with throughput for any given product mix.

We conceptualize a wafer fab as a network of queues into which products are released in a stationary (perhaps even deterministic) fashion, and processing, failure and repair times are stationary stochastic processes. Let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_I)$ be the release-rate vector, where λ_i is the release rate of product type i and there are I products. Notice that $\boldsymbol{\lambda}$ will also be the steady-state throughput, provided the combined release rates are less than the system capacity. We let $\lambda^* = \sum_{i=1}^{I} \lambda_i$ denote the overall TH of all the products. To satisfy our objective of simulation on demand, we want to use simulation to build a precise metamodel for $c_i(\boldsymbol{\lambda})$, the steady-state mean cycle time of product $i = 1, 2, \ldots, I$ for any feasible release rate $\boldsymbol{\lambda}$, by running simulation experiments at a number of settings. Throughout the paper we let $c(\cdot)$ denote steady-state mean cycle time; subscripts on c indicate fixed or non-varying quantities, while the independent variables are functional arguments. From here on we drop the subscript on c to indicate product and, without loss of generality, specialize everything to product 1.

A typical approach, which has a long history in simulation research and practice, is to fit a low-order polynomial model in $\boldsymbol{\lambda}$ using least-squares regression and classical experimental design (e.g., Kleijnen 1987, Barton and Meckesheimer 2006). Unfortunately, Cheng and Kleijnen (1999) showed that this approach fails even for relatively simple queueing simulations with a single product because the CT response surface itself, as well as the variance of the CT estimator, increase explosively as the release rate approaches the system capacity. They used queueing theory to motivate more appropriate mean and variance metamodels, and fit them by optimally allocating a fixed budget of simulation effort.

The problem is further exacerbated when PM is also a factor. Hung et al. (2005) attacked this problem by using a data driven partitioning (specifically CART) of the $\boldsymbol{\lambda}$ space, attempting to identify regions over which low-order polynomials provide a good fit. With this approach, the quality of the fit depends on how well CART is able to identify such regions.

An entirely different approach is to use an interpolation-based metamodel, such as kriging (Santner et al. 2003). Although more well known for deterministic simulation, there has been substantial recent interest in adapting these methods to stochastic simulation (e.g., Kleijnen and van Beers 2004). The primary drawbacks are that it is difficult to preserve known properties of the response surface (i.e., interpolated surfaces are often bumpy) or to tune the interpolation in high-dimensional problems (e.g., large number of products I).

We build on and extend these ideas in ways that are tailored to our specific problem. For a fixed PM (i.e., fixed percentage of each product), we exploit the approach of Yang et al. (2007ab) who extend Cheng and Kleijnen (1999) to modeling moments of CT as a function of overall TH, λ^* in complex queueing networks and drive the experiment to reach a prespecified precision rather than expend a fixed budget. This allows us to fit CT-TH curves at selected product mixes. Like Hung et al. (2005), we partition the PM space, but we do so using queueing knowledge to identify the homogeneous regions. Finally, we interpolate between fitted CT-TH curves to predict CT at product mixes we did not simulate, but we use queueing physics to develop a model-based interpolation that tends to preserve properties we know the surface should have. In the next section we develop our metamodeling approach in detail.

3 Developing the Metamodel

We first introduce some key normalizations that facilitate designing our simulation experiments and interpolating the results. We then create the metamodel families, using queueing theory to motivate the functional forms. Next, the design space is partitioned using information from the capacity model. Finally, the models are fit using a novel progressive fitting

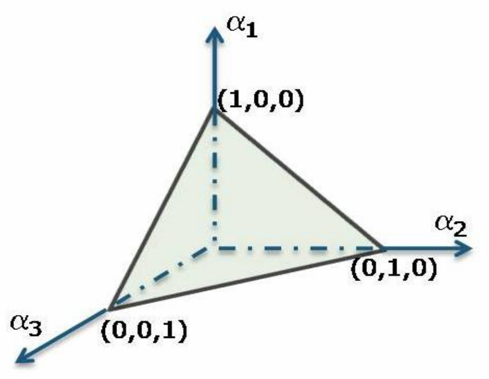

Figure 3: Mixture experiment with 3 products.

algorithm.

3.1 Key Normalizations

Our objective is to approximate the steady-state mean cycle time for all products over the entire feasible region for $\boldsymbol{\lambda}$, which is our natural design space. A point in the design space is feasible if all products have nonnegative flow and the system is stable, meaning the arrival rate to any station does not exceed the capacity of that station.

Individual product flow through the system is represented by λ_i, the release rate of product type i into the system. However, there are advantages to representing the cycle time surface as a function of λ^* and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_I)$, rather than $\boldsymbol{\lambda}$, where $\alpha_i = \lambda_i/\lambda^*$ is the fraction of type i products entering the system. Clearly, $\alpha_i \in (0,1)$ and $\sum_{i=1}^{I} \alpha_i = 1$. Given $\boldsymbol{\alpha}$ and λ^*, the individual product release rates λ_i are easily derived.

A key benefit of this representation is that experiment design over the $\boldsymbol{\alpha}$ space is a mixture experiment (Myers and Montgomery 2002). The mixture setting provides a structure for space-filling designs and has been well studied. Figure 3 illustrates the experimental region for a 3-product system, which is a simplex.

We also normalize the system throughput. A critical assumption that facilitates this is that we have a capacity model that can identify the bottleneck (BN) station (or stations) and compute its utilization as a function of release rates. Such capacity models are a standard tool in the analysis of wafer fabs; in most companies this is an internally developed spreadsheet tool, but some commercially available tools do exist (Wright, Williams, and Kelly, 2007). Although beyond the scope of this paper, the capacity calculation is often based on first solving some version of the system of flow equations, such as

$$\lambda(j) = e(j) + \sum_{\ell=1}^{J} p(\ell, j)\lambda(\ell) \qquad \forall j = 1, 2, \ldots, J$$

where J is the number of stations, $\lambda(j)$ is the overall arrival rate into station j, $e(j)$ is the arrival rate into station j from outside the system, and $p(\ell, j)$ is the fraction of products leaving station ℓ and going to station j. The utilization of station j, denoted x_j, is then computed from an equation such as

$$x_j = \frac{\lambda(j)}{s(j)\mu(j, \boldsymbol{\lambda})} \tag{1}$$

where $\mu(j, \boldsymbol{\lambda})$ is the effective service rate of a machine or resource at station j given $\boldsymbol{\lambda}$, and $s(j)$ is the number of parallel resources at station j. The utilization of the bottleneck station is therefore $x = \max_j x_j$. To incorporate machine failure and repair, batching, re-entrant flow, etc. a more sophisticated capacity model is often necessary.

We assume that for a fixed product mix $\boldsymbol{\alpha}$, the bottleneck station (or stations) does not change as a function of overall throughput λ^*. This allows us to specify system throughput in terms of x, which is also called the network "traffic intensity." To maintain positive flow of all products requires $x > 0$, while $x < 1$ guarantees stability.

Therefore, the space we want to map consists of product mixes $\alpha_1, \alpha_2, \ldots, \alpha_I$ with $\alpha_i > 0$ and $\sum_{i=1}^{I} \alpha_i = 1$, and throughput $0 < x < 1$. More specifically, we enforce $\alpha_i \geq \alpha_L$ and $x_L \leq x \leq x_U$, where α_L, x_L and x_U are user defined bounds.[1] We partition the PM space into subregions of constant bottleneck, fit CT-TH metamodels as a function of x for specific

[1] We will often ignore α_L in figures and plots since it is typically quite close to 0.

product mixes within each subregion, and interpolate between these curves as needed for new product mixes $\boldsymbol{\alpha}$ using a metamodel that is appropriate for constant-bottleneck queues. We describe this approach in more detail in the following sections.

3.2 CT-TH Curves

Yang et al. (2007a) considered the problem of estimating CT-TH curves for general queueing networks, extending the work of Cheng and Kleijnen (1999). Based on known results for tractable queueing models, heavy-traffic analysis of queueing networks, and extensive empirical study, they showed that the steady-state mean cycle time of, say, product 1 with fixed product mix $\boldsymbol{\alpha}$ could often be well approximated by a metamodel of the form

$$c_{\boldsymbol{\alpha}}(x) = \frac{\sum_{\ell=0}^{t} a_\ell x^\ell}{(1-x)^p} \qquad (2)$$

where a_ℓ, t and p are unknown parameters. They developed sequential experimental designs to place design points x and allocate experimental effort to those design points to obtain a prespecified precision across $x \in [x_L, x_U]$, the range of throughput of interest. Yang et al. (2007b) further developed this approach to simultaneously estimate the first three moment curves of steady-state CT, and to derive moment-based approximations of the percentiles of CT from them.

In the present paper, the methodology of Yang, et al. (2007a) is used to fit steady-state mean CT models of the form (2) to any desired precision at every selected PM design point. The selected PM design points constitute the design set, denoted $\mathcal{D}$. Therefore, our focus is on (a) deciding at what product mixes $\mathcal{D}$ to obtain CT-TH curves, and (b) interpolating between the available curves $\{c_{\boldsymbol{\alpha}}(\cdot), \boldsymbol{\alpha} \in \mathcal{D}\}$ to estimate mean cycle time at a specified normalized throughput, x, for product mixes that were not simulated. The interpolation in (b) is curve fitting to the data $\{c_{\boldsymbol{\alpha}}(x), \boldsymbol{\alpha} \in \mathcal{D}\}$, with a model that is motivated and presented in Section 3.3. Notice that because we normalize throughput to be $0 < x < 1$, the CT-TH curve at *every* design point $\boldsymbol{\alpha} \in \mathcal{D}$ can provide input to the interpolation at $(x, \boldsymbol{\alpha}')$ for a new product mix $\boldsymbol{\alpha}' \notin \mathcal{D}$.

3.3 A Model for CT-PM Curves

To motivate a metamodel for interpolation in the PM space, consider a multi-product M/G/1 queue. Let μ_i, $i = 1, 2, \ldots, I$ be the service rate for product type i; let σ_i^2, $i = 1, 2, \ldots, I$ be the variance of the service time of product type i; and let λ_i be the arrival rate of product type i. Then using standard M/G/1 results (e.g., Gross and Harris 1998) we can show that the steady-state mean cycle time for product 1 is

$$c(\boldsymbol{\lambda}) = \frac{1}{\mu_1} + \frac{\sum_{i=1}^{I} \lambda_i (1/\mu_i^2 + \sigma_i^2)}{2\left(1 - \sum_{i=1}^{I} \lambda_i/\mu_i\right)}. \tag{3}$$

However, if we reparameterize in terms of the product mix vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_I)$ and define the utilization to be $x = \lambda^* \sum_{i=1}^{I} \alpha_i/\mu_i$ then we can rewrite (3) as a function of $(x, \boldsymbol{\alpha})$ as

$$c(x, \boldsymbol{\alpha}) = \frac{1}{\mu_1} + \left(\frac{x}{1-x}\right)\left(\frac{\sum_{i=1}^{I} \alpha_i (1/\mu_i^2 + \sigma_i^2)}{2\sum_{i=1}^{I} \alpha_i/\mu_i}\right). \tag{4}$$

Expression (4) shows why a polynomial model is not very suitable for the CT-TH-PM surfaces of interest to us, even though it is a smooth, differentiable function.

For fixed x, the ratio $x/(1-x)$ is a constant and so (4) is a ratio-of-planes (ROP) model that can be manipulated to take the form

$$c_x(\boldsymbol{\alpha}) = \frac{\sum_{i=1}^{I} \beta_i \alpha_i}{\sum_{i=1}^{I-1} \tau_i \alpha_i + \alpha_I} = \frac{\boldsymbol{\alpha}'\boldsymbol{\beta}}{\boldsymbol{\alpha}'\boldsymbol{\tau}}. \tag{5}$$

The unknown parameters are $\boldsymbol{\beta}' = (\beta_1, \beta_2, \ldots \beta_I)$ and $\boldsymbol{\tau}' = (\tau_1, \tau_2, \ldots, \tau_{I-1}, 1)$, and they depend on the choice of x.

Of course, a semiconductor wafer fab or any nontrivial manufacturing system will consist of a number of stations, not just one. If there are J stations, and every product is processed by every station, then a plausible metamodel is

$$c_x(\boldsymbol{\alpha}) = \sum_{j=1}^{J} \left(\frac{\sum_{i=1}^{I} \beta_{ij}\alpha_i}{\sum_{i=1}^{I-1} \tau_{ij}\alpha_i + \alpha_I} \right) \tag{6}$$

the sum of J ROP models. Clearly (6) is greatly over parameterized, particularly since we want to be able to fit it by controlling $(\boldsymbol{\alpha}, x)$ and observing only estimates of the mean CTs $\{c_{\boldsymbol{\alpha}}(x), \boldsymbol{\alpha} \in \mathcal{D}\}$, rather than by modeling station-by-station performance. Further, this model is only appropriate if the bottleneck station does not shift as a function of product mix. Nevertheless, a simplified version of (6) will provide the basis for our interpolation, and motivates our choice of how to partition the experiment design space into regions of constant BN.

3.4 Partitioning the Product-Mix Space

The previous section showed that in a queueing network where a single station remains the bottleneck for all feasible product mixes, the CT response surface tends to be smooth and differentiable in $\boldsymbol{\alpha}$ over the entire region. However, if the bottleneck can shift as a function of $\boldsymbol{\alpha}$ because the products place differing loads on the stations, then the CT response surface may have a sharp, non-differentiable boundary between different bottleneck regions. We provide an illustration below.

Figure 4 is an example of a system with more than one constant-BN subregion. This system is based on a 3-product, 3-station Jackson network taken from Yang et al. (2007c).[2] In this figure the subregion where station j is the BN is denoted V_j (e.g., V_2 implies $x = x_2 = \max_{j=1,2,3} x_j$). To create a response surface in a design space with two or more BNs, we partition the product-mix space so that the bottleneck is unchanging in each subregion. When we interpolate, we only use available CT-TH curves for product mixes within the same

[2] In this example each station has a single server, and the service rate is the same for all products at a given station, but the routing of the products varies by product type. This makes the mean CT analytically tractable.

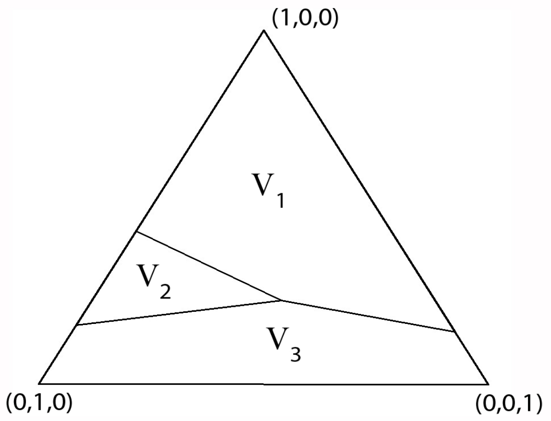

Figure 4: Constant-BN subregions shown in the product-mix space of a 3-station Jackson network.

subregion as the PM of interest. We exploit a capacity model (as described in Section 3.1) to identify the constant-BN regions.

To examine the response surface more closely for this example, we look at a cross section of the surface created by fixing $\alpha_3 = 0.1$ and letting α_1 vary from 0.0 to 0.9 (so that $\alpha_2 = 1 - 0.1 - \alpha_1$). This path through the design space is shown as a dashed line on the left plot in Figure 5. On the right in Figure 5 is a plot of the mean cycle time at $x = 0.95$, $c_{0.95}(\boldsymbol{\alpha})$, versus α_1 along this path. For this example there are 3 constant-BN subregions in the design space; within each subregion the cycle time curve is smooth and differentiable, but not at the bottleneck shifts, which motivates our choice of partition.

3.5 Progressive Fitting

Continuing the example from Section 3.4, now only consider the mean CT curves for product 2. In Figure 6 the overall CT curve for product 2 is separated into individual CT curves for each station. The figure on the left shows the whole range of α_1, while on the right only the curves for constant-BN subregion V_2 are shown. In subregions V_1 and V_3, CT for stations other than the BN increase while moving toward the BN shift. As a consequence overall CT

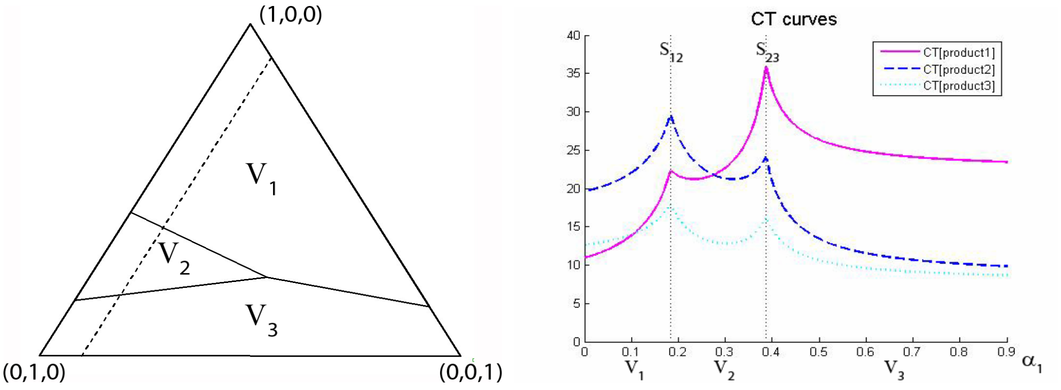

Figure 5: Product-mix path created by fixing α_3 and varying α_1 (left) and the corresponding CT-PM curves created along this path at $x = 0.95$ (right).

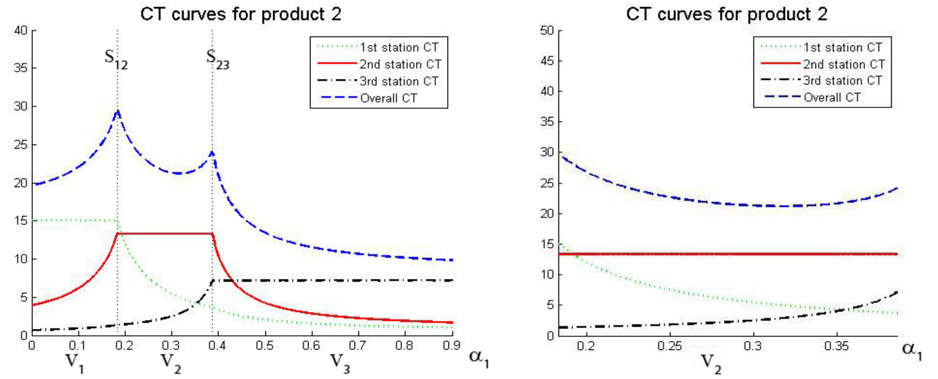

Figure 6: CT-PM curves of individual stations for product 2 over the entire region (left) and in BN subregion V_2 (right).

is monotonic. However, in subregion 2 the cycle time of station 2 stays the same, station 3's CT increases approaching V_3, and station 1's CT increases approaching V_1. As a result of being surrounded by two BN shifts, the overall CT curve in constant-BN subregion 2 is influenced greatly by two different stations' CT curves.

Even though we cannot extract individual station curves in a real problem, we want a model that can represent the sort of behavior observed in this example. Thus, in a subregion surrounded by b BN shifts, we propose progressively fitting b ROP models, one dedicated to each of the BN shifts. To do so we will utilize a weighted least squares (WLS) approach to emphasize fitting the points closer to each BN shift well. "Progressive fitting" (see Appendix A.2) implies fitting to the error surface remaining from previous fits. After fitting b ROP models, each one specialized to capture the behavior of the response surface near to one of the BN shifts, we fit one last ROP model using ordinary (unweighted) least squares.

More precisely, in a multiproduct wafer fab we approximate the expected cycle time for, say, product 1, for fixed x, in a constant-BN subregion bordered by b BN shifts, with the following model:

$$c_x(\boldsymbol{\alpha}) \approx \sum_{k=1}^{b+1} \left(\frac{\sum_{i=1}^{I} \beta_{ik}\alpha_i}{\sum_{i=1}^{I-1} \tau_{ik}\alpha_i + \alpha_I} \right) = \sum_{k=1}^{b+1} \frac{\boldsymbol{\alpha}'\boldsymbol{\beta}_k}{\boldsymbol{\alpha}'\boldsymbol{\tau}_k}. \tag{7}$$

This approach reduces the total number of parameters to $(b+1)(2I-1)$. To mitigate the chances of overfitting we use the progressive fitting scheme so that only $2K-1$ parameters are fit at a time, and we perform an overfitting test described in Appendix A.6.

We use Model (7) in two ways: During the design phase, when we build up the PM's in $\mathcal{D}$, we employ the model with $x = x_U$ to decide when enough design points have been obtained. We use $x = x_U$ because it tends to be the most difficult level of throughput to fit. Once the design set $\mathcal{D}$ is fixed, we employ interpolation based on Model (7) each time there is a query for a pair $(x, \boldsymbol{\alpha}')$ with $\boldsymbol{\alpha}' \notin \mathcal{D}$.

4 Algorithm

In this section we present a high-level description of our method for building up the design set $\mathcal{D}$ and then delivering predictions of steady-state mean CT on demand. We do so in the form of a high-level algorithm with technical details relegated to the Appendix.

The high-level algorithm begins by fitting CT-TH curves using the method of Yang et al. (2007ab) for enough product mixes that we can interpolate CT for other product mixes not simulated using these curves. We fit the CT-TH curves to very high precision, as measured by relative error across the curve, and therefore treat them as "the truth" in the interpolation steps. To judge the adequacy of our PM design set we check the goodness of fit at the most extreme TH of interest x_U, since that is typically the hardest to fit.

In our algorithm, "design points" refers to PM settings $\boldsymbol{\alpha}$; the Yang et al. (2007ab) procedure adaptively chooses throughputs x for each CT-TH curve we fit. Let $\mathcal{D}_j$ denote the set of design points supporting the interpolation metamodel for constant-BN subregion j, V_j. Without loss of generality we number the stations that could become the bottleneck as $j = 1, 2, \ldots, B$; in the worst case $B = J$, the number of stations, but in real applications B is typically 1 to 3. If two or more stations are simultaneously the bottleneck for some PMs, then we treat them as a single station for the purpose of describing the algorithm. Design points are chosen from a finely spaced grid of potential design points; let $\mathcal{P}_j$ denote the set of potential design points in V_j. We enforce $\mathcal{D}_j \cap \mathcal{P}_j = \emptyset$ so that a design point is either supporting a model or is a potential design point to be added, but not both.

Design-Phase Algorithm

Initial Tasks:

Step A: Partition the experimental region into constant-BN subregions $V_1, V_2, \ldots, V_B$ (see Section 3.4).

Step B: Select the initial set of design points for each subregion $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_B$. (see Appendix A.1).

Step C: Using the Yang et al. (2007ab) procedure, fit $c_{\boldsymbol{\alpha}}(x)$, $\forall \boldsymbol{\alpha} \in \mathcal{D}_j$, $j = 1, 2, \ldots, B$ and for each product to the selected precision level (see Section 3.2).

Main Loop: In each constant-BN subregion V_j, $j = 1, 2, \ldots, B$, perform the following until the stopping criterion is achieved. [Note: The stopping criterion is based on a user selected precision level for relative error of the metamodels. A metamodel for a given product in a given subregion is said to be "precise" if it meets the specified precision level. For the stopping criterion to be met the metamodels for all products in all subregions must be precise. See Appendix A.5. The user may also specify a maximum number of design points, D_{max}, allowed for $\mathcal{D}_j$ which acts as a secondary stopping criterion for cases where computation time must be limited.]

Step 1: Identify the products for which there is no precise metamodel and include them in a set UNFIT; the remaining products belong to the set FIT. If the set UNFIT is empty then proceed directly to Step 4.

Step 2: Do the following steps for every product labeled UNFIT.

Step 2a: Fit a single ROP model (5). If the resulting model is precise then classify the product as FIT and begin again at Step 2 for the next product. If not precise, then proceed.

Step 2b: Fit a metamodel using progressive weighted least squares (see Appendices A.2–A.3). If the metamodel is precise then proceed to Step 2c, otherwise go to Step 2d.

Step 2c: Perform a test to detect overfitting (see Appendix A.6). If there is no overfitting problem then classify the product as FIT and begin at Step 2 for the next product.

Step 2d: Find the candidate design points in $\mathcal{P}_j$ for which the relative difference between a pseudosurface (see Appendix A.4) and the fitted surface for each product in UNFIT is the greatest. Add these design points to the set $\mathcal{D}_j$ and fit $c_{\boldsymbol{\alpha}}(x)$ for each product using the procedure of Yang et al. (2007ab). [Note: The pseudosurface is a standard interpolation model that helps identify locations for new design points; see Appendix A.4.]

Step 3: Verify that each product labeled FIT is still classified as precise for all design points that now constitute $\mathcal{D}_j$. If a metamodel fails to be precise at any of these newly added points then classify it as UNFIT. Go back to Step 1.

Step 4: If $|\mathcal{D}_j| \geq D_{max}$ then the loop terminates. Otherwise, select $b + 1$ additional design points from $\mathcal{P}_j$ using a space-filling criterion. These points are used for validation of the models. Fit $c_{\boldsymbol{\alpha}}(x)$ for each product using the procedure of Yang et al. (2007ab) for these points. If the fitted model predictions at these points are precise for each product then the loop terminates. If, however, a metamodel fails to be precise then classify it as UNFIT and go back to Step 1.

At the end of this procedure we have a model structure consisting of design points $\mathcal{D} = \cup_{j=1}^{B} \mathcal{D}_j$ and CT-TH curves $c_{\boldsymbol{\alpha}}(x)$ for each product and each product mix $\boldsymbol{\alpha} \in \mathcal{D}$. To answer a query about the CT for, say, product i at $(x, \boldsymbol{\alpha}')$, we first identify to which constant-BN subregion $\boldsymbol{\alpha}'$ belongs, say j; we then use progressive fitting to fit a model of the form (7) using data $\{c_{\boldsymbol{\alpha}}(x), \boldsymbol{\alpha} \in \mathcal{D}_j\}$ and plug in $\boldsymbol{\alpha}'$ to obtain the desired result.

5 Illustration

We demonstrate our algorithm with the modified Minifab model (El Adl et al., 1996). This model features the main aspects of semiconductor manufacturing including batching, parallel machines, sequence dependent setups, recurrent flows, and stochastic tool break downs. It has three stations (five machines) and three products that share the same route, yet have different processing times. Those products will be released into the fab by three Poisson processes representing a particular product mix and TH configuration. Figure 7 shows the common processing flow for the three products.

The station families have the following characteristics:

Station Family 1 (A, B) has two parallel machines, each of which can process batches of up to 3 lots in the same processing stage following a greedy batching policy (i.e., it does not wait to fill up a batch). The two machines have independent exponentially

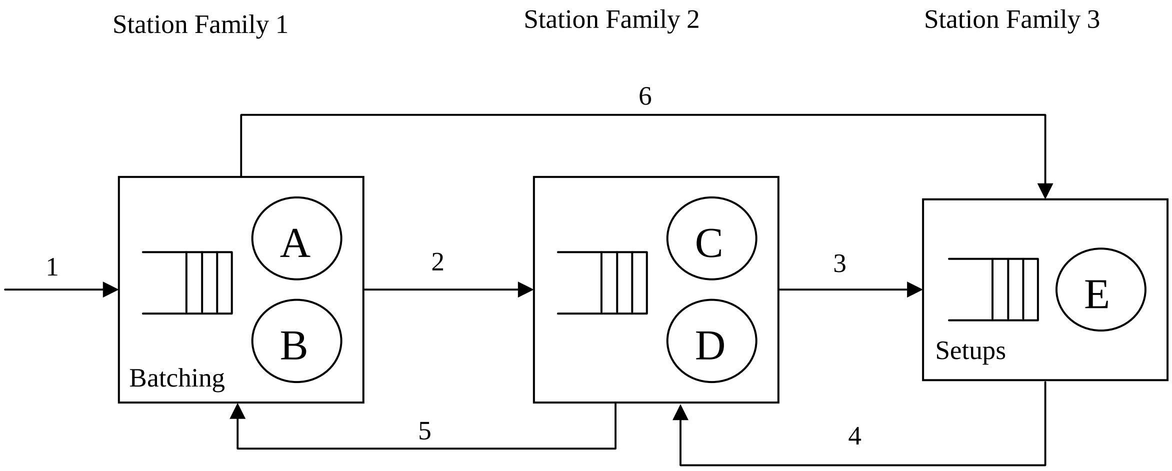

Figure 7: Processing flow in the Minifab model.

distributed time to failure and time to repair with mean values $MTTF = 29$ hours and $MTTR = 1$ hour.

Station Family 2 (C, D) also has two parallel machines. The two machines also have independent exponentially distributed time to failure and time to repair with mean values $MTTF = 9$ hours and $MTTR = 1$ hour.

Station Family 3 (E) a single machine that requires a setup of 9 minutes whenever it changes to a different processing stage than the stage currently being processed. Time to failure and time to repair are exponentially distributed with mean values $MTTF = 18$ hours and $MTTR = 2$ hours.

Table 1 shows the processing times of the three products. This set of processing times generates three regions of constant bottleneck. Product 1 has station family 3 as bottleneck, product 2 has station family 2 as bottleneck, and product 3 has station family 1 as bottleneck. The following capacity model has been derived:

$$\mathrm{Cap}_{\mathrm{StnFam\_1}}(\alpha) = \frac{1}{102.9866\alpha_1+102.9866\alpha_2+102.9866\alpha_3}$$

$$\mathrm{Cap}_{\mathrm{StnFam\_2}}(\alpha) = \frac{1}{94.9131\alpha_1+141.0137\alpha_2+73.2186\alpha_3}$$

Table 1: Processing times in minutes for the three products.

Product 1	First Visit	Second Visit	Load/Unload per Visit
Station Family 1	225	255	60
Station Family 2	50	65	30
Station Family 3	50	5	20
Product 2	First Visit	Second Visit	Load/Unload per Visit
Station Family 1	225	255	60
Station Family 2	95	105	30
Station Family 3	60	10	20
Product 3	First Visit	Second Visit	Load/Unload per Visit
Station Family 1	225	255	60
Station Family 2	30	45	30
Station Family 3	35	5	20

$$\mathrm{Cap}_{\mathrm{StnFam\_3}}(\alpha) = \frac{1}{107.33\alpha_1+123.78\alpha_2+89.89\alpha_3}$$

specifying the station family's capacity in lots per minute for some product mix $\boldsymbol{\alpha}$.

The region of interest for this example is given by minimal product percentage constraints of $\alpha_L = 0.07$ for each product and by the throughput range of $x_L = 0.5$ to $x_U = 0.95$. The simulation model is configured to collect $200,000$ CT observations after having truncated $100,000$ observations from the transient phase. Together with the minimal product percentage constraints the capacity model generates the three constant bottleneck regions shown in Figure 8.

The proposed cRSM methodology provides a MS in about 35 minutes based on the capacity model, minimal product percentage constraints, traffic intensity range, and a target of 6.5% for the relative prediction error. At each of the PM points in the design our CT-TH curve fitting algorithm (Section 3.2) decides the TH levels to simulate and allocates about 17–25 simulation replications to them until a 5% relative error criterion is achieved. In the final design we have $|\mathcal{D}| = 39$ PM points, each having simulation replications at 5 TH levels. In total, 714 simulation replications were executed. Figure 9 shows the final design.

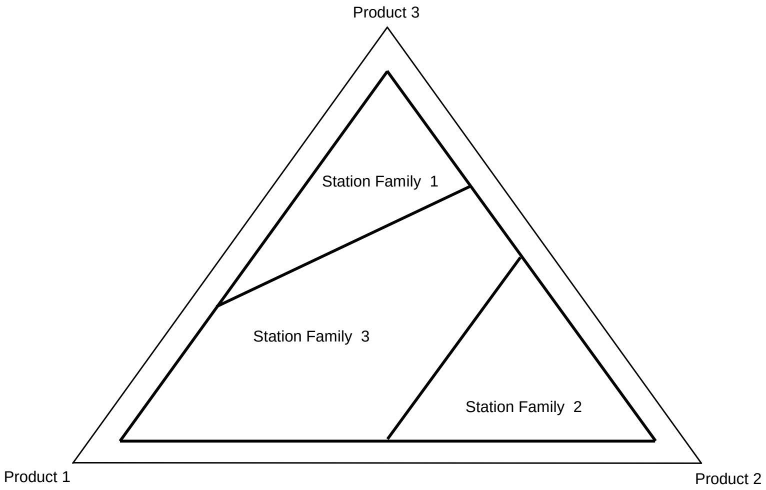

Figure 8: Product mix regions of constant bottleneck.

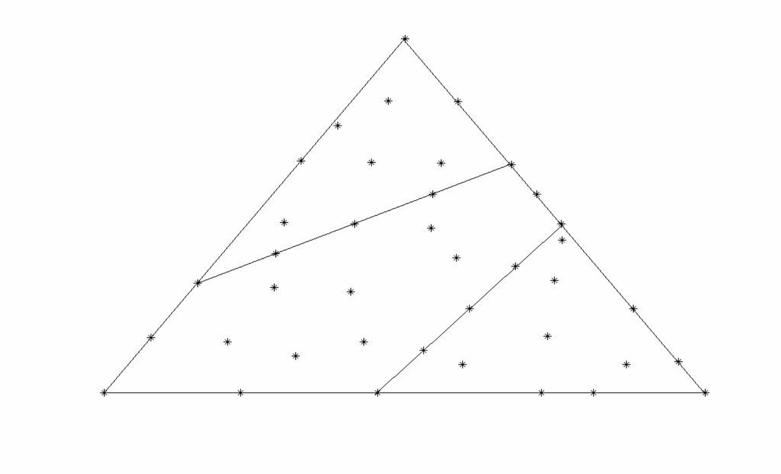

Figure 9: Final set of design points.

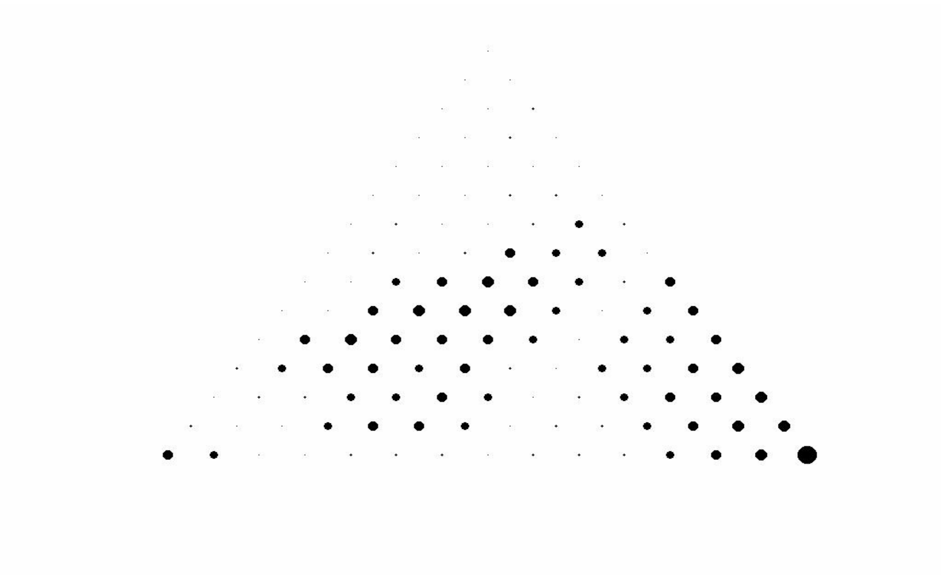

Figure 10: Relative prediction error for the mean cycle time of product 1. The largest dots represent an error between 5% to 6%. The smaller dots represent the smaller error intervals with 1%-width.

Since the true mean cycle time response surface for this Minifab model is unknown, we generate reference values for the $x = 0.95$ on a uniform PM grid of step size 0.05 via simulation using 50 replications at each grid point. These 120 values provide mean CT estimates with a confidence interval half width of less than 2% relative error.

Figures 10–12 show the relative errors of the model structure's predicted mean product cycle time in comparison to the reference values. The maximal observed error was about 6.5%, similar to the configured target relative error. Larger error values are shown in those figures around the pure mix for product 2. This appears to be caused by a certain deficiency of the capacity model in that area. Also, decreasing the target prediction error below the level 6.5% did not decrease the errors in that region.

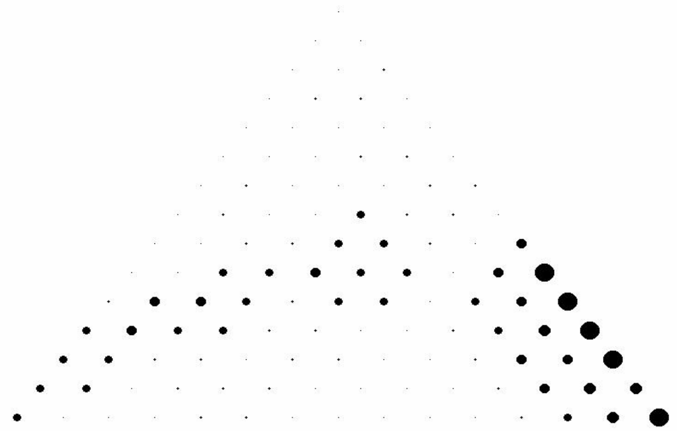

Figure 11: Relative prediction error for the mean cycle time of product 2. The largest dots represent an error between 5% to 6%. The smaller dots represent the smaller error intervals with 1%-width.

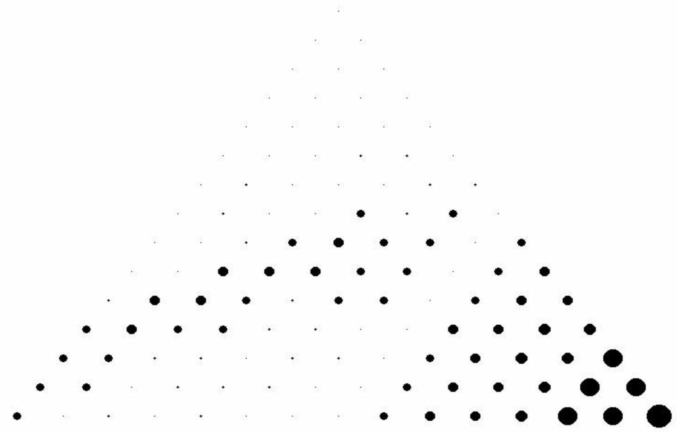

Figure 12: Relative prediction error for the mean cycle time of product 3. The largest dots represent an error between 6% to 7%. The smaller dots represent the smaller error intervals with 1%-width.

6 Conclusions

This paper has provided a methodology that performs a limited set of simulation runs for a complex manufacturing system and then uses the results of those runs to develop metamodels that predict mean steady-state cycle time as a function of product mix and throughput. These predictions can be made "on demand," i.e., without performing any additional simulation runs, for product mixes and throughput levels not previously simulated.

While this paper focused on the mean cycle time, a similar approach to predict higher moments of cycle time is being developed which will allow prediction of percentiles of cycle time. In addition, extensive empirical evaluation of the approach is currently being conducted.

Acknowledgements

This work was partially supported by Semiconductor Research Corporation Grant Numbers 2004-OJ-1224 and 2004-OJ-1225.

References

Aurand, S. and P. Miller. 1997. The operating curve: A method to measure and benchmark manufacturing line productivity. IEEE/SEMI Advanced Semiconductor Manufacturing Conference, 391–397.

Barton, R. R. and M. Meckesheimer. 2006. Metamodel-based simulation optimization. Chapter 18 in *Elsevier Handbooks in Operations Research and Management Science: Simulation* (eds. S. G. Henderson and B. L. Nelson), Elsevier.

Bekki, J. E., G. Mackulak, J. W. Fowler and B. L. Nelson. 2007. Indirect cycle time quantile estimation using the Cornish-Fisher expansion. Under review for *IIE Transactions*.

Boebel, F. G. and O. Ruelle. 1996. Cycle time reduction program at ACL. *Proceedings of the IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, 165–168.

Cheng, R. C. H. and J. P. C. Kleijnen. 1999. Improved design of queueing simulation experiments with highly heteroscedastic responses. *Operations Research* **47**, 762–777.

Chung S.-H. and C.-M. Lai. 2006. Job releasing and throughput planning for wafer fabrication under demand fluctuating make-to-stock environment. *International Journal of Advanced Manufacturing Technology* **31**, 316–327.

Cressie, N. A. C. 1993. *Statistics for Spatial Data.* Wiley, New York.

Delp, D., J. Si, Y. Hwang, B. Pei and J. Fowler. 2005. Availability adjusted X-factor", *International Journal of Production Research*, **43:18**, 3933–3953.

Delp, D., J. Si and J. Fowler. 2006. The development of the complete X-factor contribution measurement for improving cycle time and cycle time variability, *IEEE Transactions on Semiconductor Manufacturing* **13:3**, 352–362.

El Adl, M. K., A. A. Rodriguez and K. S. Tsakalis. 1996. Hierarchical modeling and control of re-entrant semiconductor manufacturing facilities. *Proceedings of the 35th Conference on Decision and Control*, Kobe, Japan.

Fowler, J. W., S. W. Brown, H. Gold and A. Schoemig. 1997. Measurable improvements in cycle-time constrained capacity. *Proceedings of the 6th International Symposium on Semiconductor Manufacturing*, San Francisco, CA, A21–A24.

Fowler, J. W., G. T. Mackulak, B. E. Ankenman, and B. L. Nelson. 2005. Procedures for efficient cycle time-throughput curve generation, *Proceedings of the NSF 2005 DMII Grantees Conference*, 1–8.

Fowler, J. W., S. Park, G. T. Mackulak and D. L. Shunk. 2001. Efficient cycle time-throughput curve generation using a fixed sample size procedure. *International Journal of Production Research,* **39:12**, 2595–2613.

Fowler, J. W., D. T. Phillips and G. L. Hogg. 1992. Control of multiproduct bulk server diffusion/oxidation processes. *IIE Transactions* **24:4**, 84–96.

Fowler, J. W., G. T. Mackulak, B. L. Nelson and B. Ankenman. 2007. Multi-product cycle time and throughput evaluation via simulation on demand. *Proceedings of the 2007 INFORMS Simulation Society Research Workshop*, Fontainebleau, France. Available via `<http://www.insead.edu/issrw/documents/08.pdf>`. [Accessed September 12, 2007].

Gross, D. and C. M. Harris. 1998. *Queueing Theory.* 3rd edition. Wiley, New York.

Hopp W. J. and M. L. Spearman. 2000. *Factory physics.* Irwin McGraw-Hill, Boston.

Hung, Y. C., G. Michailidis and D. Bingham. (2005), A framework for designing efficient simulations for complex queuing models. *Performance Evaluation*, forthcoming.

Kishimoto, M. and K. Ozawa. 2000. Optimized CMP operation by extended X-factor theory including offline unit hour", *Proceedings of the Ninth International Symposium on Semiconductor Manufacturing (ISSM)*, 71–74.

Kishimoto, M., K. Ozawa and D.P. Martin. 2001. Optimized operations by extended X-factor theory including unit hours concept, *IEEE Trans. on Semiconductor Manufacturing,* **14:3**, 187–195.

Kleijnen, J. P. C. 1987. *Statistical Tools for Simulation Practitioners.* Marcel Dekker, New York.

Kleijnen, J. P. C. and W. C. M. van Beers. 2004. Robustness of Kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *European Journal of Operational Research* **165**, 826–834.

Leachman, R. C. 2002. Competitive semiconductor manufacturing study: final report on findings from benchmarking eight-inch, sub-350nm wafer fabrication lines, University of California, Berkeley.

Leonovich, G. 1994. An approach for optimizing WIP/cycle time/output in a semiconductor fabricator, *1994 IEEE/CPMT International Electronics Manufacturing Technology Symposium.*

Mackulak, G. T., J. W. Fowler, S. Park and J. E. McNeill. 2005. A three phase simulation methodology for generating accurate and precise cycle time-throughput curves", *International Journal of Simulation and Process Modeling*, **1:1/2**, 36–47.

Martin, D. P. 1998. The advantages of using short cycle time manufacturing (SCM) instead of continuous flow manufacturing (CFM). *1998 IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, Boston, MA, September 1998, 43–49.

Martin, D. P. 1999. Capacity and cycle time –throughput understanding system (CAC-TUS): an analysis tool to determine the components of capacity and cycle time in a semiconductor manufacturing line. *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, 127–131.

Martin, D. P. 2000. Maximizing productivity improvements using short cycle time manufacturing (SCM) concepts in a semiconductor manufacturing line. *IEEE 2000 Advanced Semiconductor Manufacturing Conference (ASMC '00)*, 63–67.

Martin, D. P. 1994. Key factors in designing a manufacturing line to maintain tool utilization and minimize turnaround time. *IBM Technology Products*, Essex Junction, VT.

Myers, R. H., and D. C. Montgomery. 2002. *Response Surface Methodology: Process and Product Optimization Using Designed Experiment.* 2nd edition. Wiley-Interscience.

Nelder, J. A. and R. Mead. 1964. A simplex method for function minimization. *The Computer Journal* **7**, 308–313.

Nemoto K., E. Akcali and R. Uzsoy. 2000. Quantifying the benefits of cycle time reduction in semiconductor wafer fabrication. *IEEE Transactions on Electronics Packaging Manufacturing*, **23:1**, 39–47.

Occhino, T. J. 2000. Capacity planning model: the important inputs, formulas, and benefits, *2000 IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, 455–458.

Ozawa, K., H. Wada, H. and T. Yamaguchi. 1999. Optimum tool planning using the X-factor theory, *Proceedings of the 1999 IEEE International Symposium on Semiconductor Manufacturing (ISSM)*, 49–52.

Park, S., J. W. Fowler, G. T. Mackulak, J. B. Keats and W.M. Carlyle. 2002. D-Optimal sequential experiments for generating a simulation-based cycle time-throughput curve, *Operations Research*, **50:6**. 981–990.

Pfund, M. E., S. J. Mason and J. W. Fowler. 2006. Semiconductor manufacturing scheduling and dispatching, *Handbook of Scheduling, edited by J.W. Herrmann, Springer International Series*, 213–242.

Piepel, G. F. 1988. Programs for generating extreme vertices and centroids of linearly constrained experimental regions. *Journal of Quality Technology* **20**, 125–139.

Santner, T. J., B. J. Williams and W. I. Notz. 2003. *The Design and Analysis of Computer Experiments.* Springer, New York.

Schoemig, A. K. 1999. On the corrupting influence of variability in semiconductor manufacturing, *Proceedings of the 1999 Winter Simulation Conference (WSC)*, 837–842.

Semiconductor Industry Association. 2006. International Technology Roadmap for Semiconductors 2006 update, `<http://www.itrs.net/Links/2006Update/FinalToPost/10_Factory_2006Update.pdf>`. [Accessed January 2007].

Simpson, T. W., and F. Mistree. 2001. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal* **39**, 2233–2241.

Spence, A.M. and D.J. Welter. 1988. Capacity planning of a photolithography work cell in a wafer manufacturing line, *Proceedings of the IEEE International Conference on Robotics and Automation*, Raleigh, NC, 702–708.

Wright, Williams, and Kelly, 2007. Factory Explorer Users' Guide.

Yang, F., B. E. Ankenman and B. L. Nelson. 2007a. Efficient generation of cycle time-throughput curves through simulation and metamodeling. *Naval Research Logistics* **54**, 78–93.

Yang, F., B. E. Ankenman and B. L. Nelson. 2007b. Cycle time percentile curves for manufacturing systems. *INFORMS Journal on Computing*, forthcoming.

Yang, F., J. Liu, B. L. Nelson, B. E. Ankenman and M. Tongarlak. 2007c. Metamodeling for cycle time-throughput-product mix surfaces using progressive model fitting. Working paper, Industrial and Management Systems Engineering, West Virginia University.