

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Stochastics and Statistics

Neural network metamodeling for cycle time-throughput profiles in manufacturing

Feng Yang*

Industrial and Management Systems Engineering Dept., West Virginia University, P.O. Box 6070, Morgantown, WV 26506, United States

ARTICLE INFO

Article history:

Received 30 October 2008

Accepted 31 December 2009

Available online 11 January 2010

Keywords:

Discrete event simulation

Response surface modeling

Design of experiments

Neural networks

Semiconductor manufacturing

Queueing

ABSTRACT

This paper proposed a neural network (NN) metamodeling method to generate the cycle time (CT)–throughput (TH) profiles for single/multi-product manufacturing environments. Such CT–TH profiles illustrate the trade-off relationship between CT and TH, the two critical performance measures, and hence provide a comprehensive performance evaluation of a manufacturing system. The proposed methods distinct from the existing NN metamodeling work in three major aspects: First, instead of treating an NN as a black box, the geometry of NN is examined and utilized; second, a progressive model-fitting strategy is developed to obtain the simplest-structured NN that is adequate to capture the CT–TH relationship; third, an experiment design method, particularly suitable to NN modeling, is developed to sequentially collect simulation data for the efficient estimation of the NN models.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

This paper is concerned with the steady-state performance modeling of a single/multi-product manufacturing system. The performance metrics of primary interest are the throughput (TH) and manufacturing cycle time (CT). TH is defined as the rate at which entities are processed by the system, and CT refers to the time it takes an entity to traverse the system (Hopp, 2007). The trade-off relationship between CT and TH, which is the subject of this paper, has long been recognized to provide a comprehensive performance profile of a manufacturing system (Atherton and Dayhoff, 1986; Hopp and Spearman, 2008). Decision makers can use such CT–TH profiles to compare the manufacturing efficiency of different system configurations (defined in terms of the number of equipments, number of operators, etc.), and hence improve the managerial decisions such as long-term capacity expansion and staffing. Fig. 1 gives an example of CT–TH profiles for single-product systems; each curve describes the performance of a different system configuration. We refer the interested readers to Spence and Welter (1987) for discussions of selecting the best system configuration among different scenarios based on their CT–TH profiles.

Because of the critical role played by CT–TH profiles in characterizing system performance, substantial research effort has been devoted to quantifying this trade-off relationship for manufacturing systems. The majority of existing methods can be divided into two categories: queueing theory and computer simulation. A recent review of queueing models for manufacturing systems is

given in Shantikumar et al. (2007), which includes Whitt (1983), Bitran and Tirupati (1988), Chen et al. (1988), Mitrani and Puhalskii (1993), Connors et al. (1996), Morrison and Martin (2007), etc. Queueing models, though fast and easy to use, rely on restrictive assumptions and may not be able to accurately capture the CT–TH relationship for real complex systems. Discrete event simulation, on the other hand, is an alternative approach which can include any details that are important to the system. Because of its high fidelity and flexibility, and increasingly also because of its ease of use, simulation has become an essential tool for decision making in manufacturing, which is especially true in the semiconductor industry (see, for instance Schömig and Fowler, 2000), the motivating application for this research. However, simulation may be very time-consuming to run: models of complex manufacturing systems may take several hours for a single replication (Fowler and Rose, 2004). Moreover, simulation merely provides a means to evaluate the CT of entities with given input parameters, which includes the system configurations as well as the TH (equivalent to the release rate of entities into the system in steady-state); that is, a simulation run can only provide an estimated point in the CT–TH performance space, and plus, many replications may be required to achieve a good estimation. Hence, simulation, by itself, is a clumsy tool to explore the CT–TH relationship, and the computational cost involved may well hinder the decision makers from adequately considering the trade-off characteristics of system performance.

Aiming at overcoming the major drawbacks of queueing methods and computer simulation, the author and her co-authors (Yang et al., 2007, accepted for publication) recently proposed a metamodeling approach for the efficient generation of CT–TH profiles.

* Tel.: +1 304 293 4607x3714.

E-mail address: Feng.yang@mail.wvu.edu

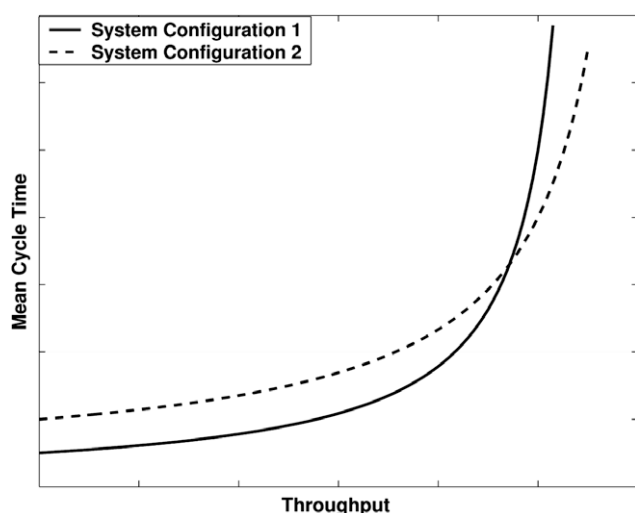


Fig. 1. CT-TH profiles for two different system configurations.

A metamodel, which takes the form of polynomial regressions, splines, etc., is a mathematical approximation of the quantitative relationship implied by the simulation. Metamodeling techniques refer to the integration of computer simulation and response surface modeling (Chapter 18, Henderson and Nelson, 2006). Specifically, to metamodel the CT-TH profiles (TH and CT) data pairs will first be collected by running a selected set of simulation experiments; based on the data, statistical methods will be used to fit a metamodel representing the CT-TH relationship. The resulting metamodel is a mathematical function like that provided by a tractable queueing model while possessing the high fidelity of simulation. Efficient metamodeling of the CT-TH profiles is not easy due to the special features of the performance surfaces; this has been illustrated in Yang et al. (2007, accepted for publication) and will be reiterated where appropriate in this paper. Assuming a single-product environment, Yang et al. (2007) developed a metamodeling procedure to generate CT-TH curves (Fig. 1). Built upon and substantially extended beyond the previous work, Yang et al. (accepted for publication) characterized the CT-TH surfaces for multi-product systems where TH refers to the throughput vector with each element representing the output rate of a certain type of products. In both papers, traditional nonlinear regression metamodels were adopted for the CT-TH profiles. Although demonstrated effective through extensive empirical evaluation, the methods developed in these works have two shortcomings:

- The traditional regression-based metamodeling involves very sophisticated experiment design and nonlinear fitting strategies, and hence is difficult to implement in industry.
- The nonlinear regression models fall short in capturing the CT-TH surfaces over a relatively wide range of TH input. This will become clearer in Section 5.3.

To address these shortcomings, we proposed in this paper a neural network (NN)-based metamodeling approach, which is easy to implement in practice, and able to efficiently generate high-quality CT-TH profiles over a wide TH range for both single and multi-product environments. On the methodology side, we utilized the geometry of NN and developed efficient statistical methods to achieve well-estimated NN models. Although NN is widely used as a powerful metamodeling tool in manufacturing (Vellido et al., 1999; Sabuncuoglu and Touhami, 2002), building a successful NN model remains a challenging task due to the various difficulties discussed in Curry and Morgan (2006) and Zhang (2007). The majority of existing work simply feeds the data to commercial

softwares such as NeuralWorks for NN training, and rarely considers the issues of statistical validity or computational efficiency. Our methods distinct from the large amount of NN modeling work in three aspects. (i) The geometrical congeniality is recognized between the single-hidden layer feedforward neural networks (SLFNs), the selected NN metamodel in this paper, and the target performance surfaces. Such congeniality endows the SLFN with the potential to well approximate the CT-TH profiles (Section 3.2). In the literature, NN has largely been treated as a black box, whereas the geometry of nonlinear models, to which NN belongs, is known to play an important role in response surface modeling (Seber and Wild, 2003). In our methods, the geometry of SLFN was utilized to achieve high-quality CT-TH surfaces. (ii) An iterative model-fitting strategy was developed to obtain an SLFN of the simplest form and that can accurately capture the CT-TH profile (Section 3.3). Model selection, i.e., identifying the simplest NN topology that is sufficient to characterize the sample data, is a difficult task. Existing NN modeling work rarely takes any steps beyond the “rule of thumb” to select the appropriate model complexity. Here, a singular value decomposition based method is proposed to search for the best fitted NN model with the smallest number of hidden nodes. (iii) A sequential experiment design procedure is developed to run simulation experiments efficiently for the fitting of the NN model (Section 4). To the best of our knowledge, this is the first attempt to develop a sequential design strategy that is particularly suitable for NN modeling.

The remainder of this paper is organized as follows. Section 2 defines the research problem in precise terms and provides an overview of the proposed method. Section 3 discusses the technical details of NN-based response surface modeling. In Section 4, a complete multi-stage procedure is given for estimating the CT-TH performance surfaces via sequential simulation experiments. In Section 5, an empirical evaluation of the proposed NN metamodeling is given, and a comparison to the traditional regression-based approach is performed.

2. Background of the CT-TH modeling

The objective of this paper is to develop an NN-based metamodeling approach for the generation of CT-TH profiles that capture the comprehensive performance for a manufacturing system. Since a single-product environment is a special case of multi-product systems, we focus on the generation of CT-TH surfaces for multi-product systems, which is also the research subject of Yang et al. (accepted for publication). The superiority of the proposed methods over Yang et al. (accepted for publication) has been briefly mentioned in Section 1 and will also be discussed in details later. To state the research problem in more precise terms, we define some notation here. The product flow of a manufacturing system is described as follows:

- λ : the overall release rate of all the products into the system.
- $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_K)$: the product-mix (PM) vector with each element α_k representing the fraction of type k products in the flow; we have $\sum_{k=1}^K \alpha_k = 1$, $\alpha_k \in [0, 1]$.
- $\lambda_k = \alpha_k \lambda$: the release rate of type k products to the system.

Assuming steady-state, the TH is equivalent to the release rate of products into the system, and is fully specified by either (λ, α) or equivalently, $(\lambda_1, \lambda_2, \dots, \lambda_K)$. CT, technically, is a random variable representing the total time required for an entity to traverse the system, and the steady-state CT distributions depend on the system TH.

We denote the target CT-TH relationship by $c_k(\lambda, \alpha)$, which quantifies the dependence of the mean CT for type k products ($k = 1, 2, \dots, K$) upon the throughput flow (λ, α) . In the remainder

of this paper, (λ, α) will be referred to as the input variables of the CT–TH surface; and the mean CT will be regarded as the response of the target surface. For conciseness, we will use CT to refer to mean CT without causing any confusion. Given a simulation model representing the manufacturing system of interest, simulation runs will be performed at selected points in the (λ, α) input region to collect data, from which the performance surfaces $\{c_k(\lambda, \alpha), k = 1, 2, \dots, K\}$ can be estimated.

In the remainder of this section, we briefly outline the analytical analysis involved in generating CT–TH profiles, which is relatively independent of the NN metamodeling methods (Sections 3 and 4), the focus of this paper. The basic ideas presented in Sections 2.1, 2.2 and 2.3 were initially proposed in Yang et al. (accepted for publication), and are provided here for the sake of completeness and to give the reader more insights into the target response surface.

2.1. Preliminary queueing analysis

Our primary research focus is on generating the CT–TH surfaces $\{c_k(\lambda, \alpha), k = 1, 2, \dots, K\}$ via simulation metamodeling. However, as pointed out by Yang et al. (accepted for publication), the complex nature of the target surfaces calls for a preliminary queueing analysis which serves two purposes:

- To approximate system capacity and identify bottleneck resources, which facilitates the definition and normalization of the (λ, α) input region.
- To divide the feasible input region into a number of subregions, which allows for the fitting of a smooth response surface within each subregion.

These will become clearer in Sections 2.2 and 2.3. We represent the manufacturing system as a queueing network consisting of M stations. The notations related to capacity/bottleneck analysis are given as:

- $x \in (0, 1)$: the system utilization; $x = \rho_{max} = \max_j \rho_j$ where ρ_j is the utilization of station j ($j = 1, 2, \dots, M$).
- Bottleneck (BN) station j_{BN} : the station that reaches ρ_{max} .
- $u^*(\alpha)$: the system capacity, the upper limit on λ (or overall throughput) for stability.

Both capacity $u^*(\alpha)$ and the BN station j_{BN} depend on the system parameters as well as the PM α . As in Yang et al. (accepted for publication), it is assumed that for a given system configuration and PM, existing queueing models can be used to approximate the capacity $u^*(\alpha)$ and to identify BN station(s) for realistic manufacturing systems (with set-ups, batching, re-entrant flows, etc.) Examples of such queueing models include Hopp et al. (2002), Kumar and Kumar (2001) and Meng and Heragu (2004). For more discussions on the queueing analysis, see Appendix A.1.

The analytical queueing analysis serves as a preliminary step prior to the metamodeling of the CT–TH surfaces. The two-stage approach, i.e., queueing analysis plus metamodeling, is based on the premise that queueing models are much more accurate for capacity analysis rather than for estimating the expected cycle times (Hopp et al., 2002). The latter is handled by simulation metamodeling in our research.

2.2. The target CT–TH surfaces

As will become clearer later, after invoking the analytic engine to perform capacity/bottleneck analysis (Section 2.1), simulation will be performed to collect data for the fitting of the CT–TH surfaces $\{c_k(\lambda, \alpha), k = 1, 2, \dots, K\}$.

In the simulation experiments, (λ, α) are considered as controllable input variables. The stability condition of the system is such that λ has to be less than the system capacity $u^*(\alpha)$, which as already established, can be analytically approximated for a given product-mix α . For reasons that will become apparent in Section 2.3, instead of estimating $c_k(\lambda, \alpha)$ we normalize the range of λ across the PM region and directly estimate $c_k(x, \alpha)$. The utilization $x = \lambda/u^*(\alpha)$ is the fraction of system capacity in use, and x is on the scale of $[0,1)$ regardless of the value of α . Once we have obtained $c_k(x, \alpha)$, a simple transformation will provide $c_k(\lambda, \alpha)$. Note that the system capacity $u^*(\alpha)$ obtained from queueing analysis is what makes this transformation possible.

From now on, we will also refer to the performance surfaces as CT– x –PM surfaces, and our task is to estimate $\{c_k(x, \alpha), k = 1, 2, \dots, K\}$ for K different types of products in the system. We use Γ to denote the feasible input region, which is defined by the utilization x and PM α . As detailed in Appendix A.2, the feasible region is given by:

- $x \in [x_L, x_U]$ where $0 < x_L < x_U < 1$
- Ω , the feasible space for α , is a linear simplex defined as

$$\sum_{k=1}^K \alpha_k = 1, \quad \alpha_k \in [0, 1].$$

$$A\alpha \leq b. \tag{1}$$

where (1) are the linear constraints imposed on product-mix due to practical considerations in manufacturing.

Hence, the input region is given as $\Gamma = [x_L, x_U] \times \Omega$, the Cartesian product of set $[x_L, x_U]$ and set Ω . In Section 2.3, we discuss the shape of the CT– x –PM surface through an open Jackson network, and illustrate the partition of the input region Γ to facilitate our metamodeling.

2.3. Motivating systems

Following the notation in Appendix A.1, we consider a Jackson network in which each station has a single server having exponentially distributed service time with rate u_j (independent of product type). Given the system parameters for this network, the expected cycle time for each product type can be derived analytically as a function of PM α . Since all $c_k(x, \alpha)$ ($k = 1, 2, \dots, K$) functions have the same form we consider the cycle time of product 1 without loss of generality:

$$c_1(x, \alpha) = \sum_{j=1}^M \frac{\delta_{1j}}{u_j \left[1 - x \left(\frac{\sum_{k=1}^K \alpha_k \delta_{kj} / u_j}{\max_h \sum_{k=1}^K \alpha_k \delta_{kh} / u_h} \right) \right]} \quad x \in [x_L, x_U], \alpha \in \Omega. \tag{2}$$

In (2), all but x and α are system parameters, and a station that achieves $\max_h \sum_{k=1}^K \alpha_k \delta_{kh} / u_h$ is a BN station (see Appendix A.1 for details). Within a subregion $\Gamma_v = [x_L, x_U] \times \Omega_v$, where station v stays the BN, (2) can be written as:

$$c_1(x, \alpha) = \sum_{j=1}^M \frac{\delta_{1j}}{u_j \left[1 - x \left(\frac{\sum_{k=1}^K \alpha_k \delta_{kj} / u_j}{\sum_{k=1}^K \alpha_k \delta_{kv} / u_v} \right) \right]} \quad x \in [x_L, x_U], \alpha \in \Omega_v. \tag{3}$$

It is obvious from (3) that the cycle time is a continuous and differentiable function of x and α within a constant-BN subregion. Fig. 12 in Appendix A.4 provides a graphical illustration of the CT– x –PM surface through a simple 3-product 3-station Jackson network. This motivates us to divide the feasible PM region into a number of constant-BN subregions, and separately fit a response surface to each subregion Γ_v .

For real manufacturing systems (involving reworks, machine failures, batch processes, setups, etc.), the PM region partition can be obtained through the preliminary analytical analysis (Section 2.1) using the existing queueing models in the literature. Furthermore, our extensive empirical experience with real semiconductor fabrication models shows that, the CT- x -PM surfaces of realistic systems display the same features as those of Jackson networks.

In light of the discussions above, we follow the following meta-modeling strategy: first, preliminary queueing analysis is used to perform the PM region partition (Specifics are given in Appendix A.3), and then within each subregion, a smooth response surface is fitted for $c_k(x, \alpha)$ ($k = 1, 2, \dots, K$) with $x \in [x_L, x_U]$ and $\alpha \in \Omega_v$.

3. Metamodeling of the CT- x -PM surfaces

In this section, we perform the NN-based metamodeling for the CT- x -PM surfaces $\{c_k(x, \alpha); k = 1, 2, \dots, K\}$ within a constant-BN subregion $\Gamma_v = [x_L, x_U] \times \Omega_v$ defined in Section 2.3.

3.1. Single-hidden layer feedforward networks (SLFNs)

Fig. 2 gives an example of the structure of an SLFN, the functional form of which is given as:

$$Y(\mathbf{u}) = f(\mathbf{u}, \theta) + \varepsilon = \sum_{t=0}^T b_t \cdot h(\mathbf{w}_t' \mathbf{u}) + \varepsilon, \quad \mathbf{u} \in \Gamma_v. \quad (4)$$

We define the following notations:

- $\mathbf{u} = (1, u_1, u_2, \dots, u_K)'$: the $(K + 1) \times 1$ vector including K input factors to the network plus the constant term 1. In our CT- x -PM modeling, $\mathbf{u} = (1, x, \alpha_1, \alpha_2, \dots, \alpha_{K-1})'$ includes the utilization x and $K - 1$ linearly independent product-mixes.
- $\Gamma_v = [x_L, x_U] \times \Omega_v$: the input region of the vector \mathbf{u} . For notation convenience, we will not discriminate between vector \mathbf{u} defined above (with a constant 1) and the input variables $(x, \alpha_1, \alpha_2, \dots, \alpha_{K-1})'$. The difference between them is obvious.
- $Y(\mathbf{u})$: the simulation output obtained at input setting \mathbf{u} ; in the CT- x -PM modeling, $Y(\mathbf{u})$ represents the CT estimate at $\mathbf{u} = (x, \alpha)$ with $E[Y(\mathbf{u})] = c_k(x, \alpha)$, the steady-state mean CT for products of type k ($k = 1, 2, \dots, K$). Without loss of generality, we will focus on the fitting of $c_1(x, \alpha)$ in the remainder of this section.
- ε : error term that follows normal distribution with expectation 0 and constant variance σ^2 . As will be seen in Section 4.1, σ^2 is a user-specified parameter, and hence is considered as known in our statistical modeling. The justification for the normal assumption on ε is also given in Section 4.1.

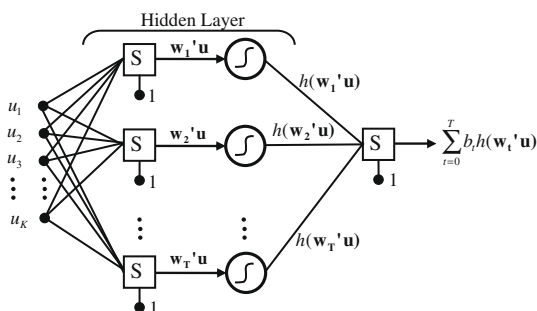


Fig. 2. An example of an SLFN.

- $h(\cdot)$: the nonlinearity of the hidden layer activation functions. In this paper, we adopted the sigmoid function:

$$h(\mathbf{w}_t' \mathbf{u}) = \frac{1}{1 + \exp(\mathbf{w}_t' \mathbf{u})}, \quad t = 1, 2, \dots, T. \quad (5)$$

In the formulation of model (4), it is assumed that $h(\mathbf{w}_0' \mathbf{u})$ is a constant 1.

- T : number of hidden neurons.
- $\mathbf{w}_t = (w_{t,0}, w_{t,1}, w_{t,2}, \dots, w_{t,K})'$ is the $(K + 1) \times 1$ weight vector for the t th hidden neuron ($t = 1, 2, \dots, T$).
- $\mathbf{b} = (b_0, b_1, \dots, b_T)$ are the weight parameters from the hidden layer to the output neuron. Note that $w_{t,0}$ and b_0 are usually referred to as bias weights.
- θ is the vector including all the network parameters, $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_T\}$ and \mathbf{c} .

Suppose that there are N simulation samples $\{(\mathbf{u}_i, y_i), i = 1, 2, \dots, N\}$. In addition, we define:

U : the $N \times (K + 1)$ matrix of input vectors with the i th ($i = 1, 2, \dots, N$) row being the i th input vector

$$\mathbf{u}_i = (1, u_{i,1}, u_{i,2}, \dots, u_{i,K}).$$

W : the $(K + 1) \times T$ matrix of weight with the i th ($t = 1, 2, \dots, T$) column being

$$\mathbf{w}_t = (w_{t,0}, w_{t,1}, w_{t,2}, \dots, w_{t,K})'.$$

$H = h(UW)$: the $N \times (T + 1)$ matrix of the hidden layer's outputs. The notation $h(Z)$ represents a map which takes a matrix Z with elements z_{ij} and returns another matrix of the same size with elements $h(z_{ij})$, where h is the neuron's nonlinearity. We have:

- The i th ($i = 1, 2, \dots, N$) row of H is $(h(\mathbf{w}_0' \mathbf{u}_i), h(\mathbf{w}_1' \mathbf{u}_i), \dots, h(\mathbf{w}_T' \mathbf{u}_i))$, representing the $T + 1$ hidden layer outputs for input \mathbf{u}_i .
- The i th ($t = 2, 3, \dots, T + 1$) column of H is a $N \times 1$ vector with the i th ($i = 1, 2, \dots, N$) element being the output from the $(t - 1)$ th hidden neuron for input \mathbf{u}_i . Notice that the 1st column of H is a constant vector $\mathbf{1}_{N \times 1}$.

3.2. Geometrical interpretation of SLFNs

It is common to treat the NN as a “black box”, and researchers frequently make no reference to the geometry of the NN, which could be very helpful in response surface modeling. Here, we briefly review the geometrical perspective provided by Xiang et al. (2005). For graphical illustration, we consider a case with two input factors $\mathbf{u} = (u_1, u_2)$. Each hidden node corresponds to a sigmoid activation function (5), which is a hypersurface as depicted in Fig. 3. These hypersurfaces can be considered as the projection of the input space onto the activation functions, and they act as the basic building blocks of the network. The position and shape of each plane depend on the weight parameters of that neuron.

It is well-known that a SLFN can adapt to, i.e., approximate, arbitrary functional forms (White, 1989). Although this “universal approximation” theorem relies on the assumption that the SLFN can include an infinite number of hidden nodes, in our metamodeling SLFNs demonstrate a good capability in approximating the CT- x -PM surfaces: based on a sample of very moderate size, a SLFN is able to provide a highly accurate approximation to the target response surface using a small number of hidden nodes (Section 5). We believe that this is at least in part, due to the particular shape of the CT- x -PM surface. Consider for example a low-dimensional case where $K = 2$

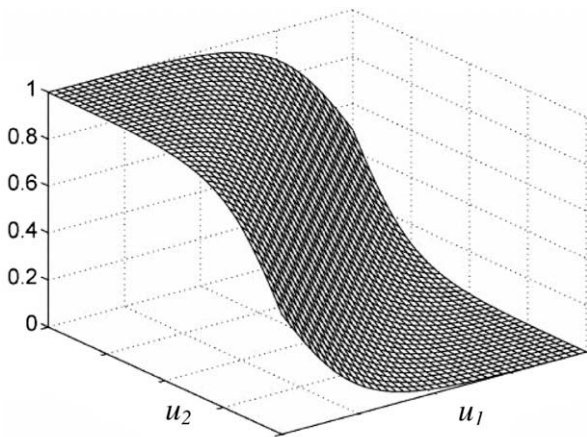


Fig. 3. 2-Dimensional sigmoid activation function.

different types of products are involved. We have $\mathbf{u} = (x, \alpha_1)$, and the surface can be denoted as $E[Y(\mathbf{u})] = c_1(x, \alpha_1)$. For simple queueing models such as open Jackson network (Section 2.3), $c_1(x, \alpha_1)$ is bowl-shaped with a flat bottom and very steep side walls. This trend of the surface is likely to be well captured by the lower half of the sigmoid activation function as plotted in Fig. 3.

Nevertheless, using a SLFN to efficiently metamodel the CT- x -TH surface is difficult. In this paper, we proposed model fitting (Section 3.3) and experiment design (Section 4) strategies to insure that a high-quality SLFN-based surface is obtained in a computationally efficient manner. It is worth emphasizing that these design and fitting methods all bear on the concept that each neuron in the hidden layer projects the input space onto a hypersurface.

3.3. Estimation of the SLFN-based response surface

Based on the sample $\{(\mathbf{u}_i, y_i), i = 1, 2, \dots, N\}$, the SLFN fitting problem is defined as

$$\min_{\theta} \text{SSE}(\theta) = \sum_{i=1}^N [y_i - f(\mathbf{u}_i, \theta)]^2 \quad (6)$$

where θ represents the vector of network parameters.

Obtaining good estimates for θ by solving the nonlinear optimization problem (6) is difficult. In our work, an iterative model-fitting strategy is developed to search for an adequate SLFN model with no redundant hidden nodes. As illustrated in Fig. 4, we start with a SLFN with $T = 1$ hidden node, and expand the current network by incorporating one additional hidden neuron at a time until a best fitted model is obtained. Next, we discuss in details the various issues involved in the fitting process.

3.3.1. Least-square fitting

Step (a) of the fitting process (Fig. 4) is to fit a SLFN model to the sample data with given model structure, i.e., fixed number of hid-

Initialization: Set $T = 1$; Enter the following iteration.

Step (a): Based on the sample data, fit the SLFN model with T hidden nodes.

Step (b): Evaluate the resulting fitted model for redundant hidden nodes.

If there is no redundancy in the T -hidden node network, set $T = T + 1$ and go to Step (a).

Otherwise, set $T = T - 1$, and declare the fitted SLFN model with T hidden nodes as the best model based on the current sample.

Fig. 4. Iterative SLFN fitting strategy.

den neurons. Like most nonlinear regression, the least-square fitting (6) is almost bound to suffer from the trap of local optima. In the literature, “global optimization” techniques such as genetic algorithm have been used in search of the global optima. In this work, no global search techniques were implemented, and Matlab Neural Network Toolbox was used to fit the SLFNs. In our empirical experience, the performance of the least-square fitting is not sensitive to the starting values of NN parameters θ . To assess the effect of starting points upon the resulting $\hat{\theta}$, for each example detailed in Section 5, NN models (with selected model structure, i.e., number of hidden nodes) were fitted using 500 randomly generated starting points. For those examples, among their 500 trials, about 5–10% of the times θ converged to a local rather than a global optima. Thus, a global optima can be largely ensured by carrying out the model fitting with a couple of tens of random starts and selecting the θ that achieves the smallest $\text{SSE}(\theta)$. Of course, the number of random starting points that may be needed to achieve a global optima depends on the dimension of the unknown parameter vector θ (or the number of hidden nodes in the NN). Our recommendation on the number of starts applies to an SLFN with 2–6 hidden nodes, which in our experience (Section 5) is usually sufficient to approximate the target CT-TH surface.

Also, it is worth mentioning that due to lack of model identification (Section 3.3.2), the least-square objective function (6) has a set of, as opposed to a single, global optima; the points in the set of global optima are considered equivalent in terms of the model fitting error. The NN model resulting from Step (a), Fig. 4 is the basis for neuron redundancy evaluation in Step (b), and thus is important to determining the appropriate number of hidden neurons, which will be detailed discussed in Section 3.3.3.

3.3.2. Model identification

Lack of model identification is a serious problem recognized in the literature for NN modeling (Anders and Korn, 1999; Curry and Morgan, 2006). For a precise definition of model identification, see Davidson and MacKinnon (1993, Chapter 2). Here, we focus our attention on local identification of the NN model, which is a necessary condition for valid statistical inference on the fitted model. Denoting $\hat{\theta}$ as the least-squares parameter estimate in (6), $\hat{\theta}$ will be locally identified if the function $\text{SSE}(\theta)$ is strictly convex at $\hat{\theta}$. For SLFNs, there are two major sources of model unidentification.

- First, as pointed out by White (1989), the presence of redundant hidden nodes certainly causes the SLFN model to be locally unidentified. If the SLFN (4) includes an irrelevant hidden neuron t , then the hidden layer weights \mathbf{w}_t will have no effect on the value of $\text{SSE}(\theta)$, and hence the strict convexity of $\hat{\theta}$ will be violated. In Fig. 4, we eliminate this hidden node redundancy by the progressive fitting strategy: initially the simplest SLFN with a single hidden neuron is fitted, and then the model complexity will be enhanced by incorporating one additional hidden node at a time until the best SLFN free of redundancy is identified.
- The second problem is more serious than the first one. Through a theoretical analysis, Curry and Morgan (2006) showed that there will almost inevitably be approximate functional relationships between the network parameters, even for very simple NN models. There is no easy way to circumvent the strong interdependency among model parameters, which remains one of the existing difficulties with NN modeling (Curry and Morgan, 2006).

The resulting SLFN from our progressive fitting process is free of the hidden node redundancy, but still may be poorly identified as any NN model in the literature. A direct consequence of lack of model identification is that the standard statistical inference

methods for nonlinear regression are no longer valid (Seber and Wild, 2003). In our work, alternative approaches are proposed to handle the statistical inference issues, particularly the model selection issues.

3.3.3. Model selection using the singular value decomposition

In the SLFN fitting, the major model selection issue is the determination of the number of hidden layer neurons T , which characterizes the complexity of the model. Our iterative fitting process aims at achieving a fitted network with the smallest T out of two reasons. First, a simplest model requires the smallest number of design points and hence the least amount of simulation. Second, as explained in Section 3.3.2, we want to eliminate the lack of model identification due to the existence of redundant hidden nodes.

As illustrated in Fig. 4, we start with a SLFN with $T = 1$ hidden neuron, fit the model with the specified structure, and then evaluate the fitted model to see if it includes redundant neurons. If the answer is YES, we delete one node from the hidden layer and stop; otherwise, set $T = T + 1$ and repeat the process. In the progressive structure, model selection amounts to deciding whether or not an hidden node should be omitted from the SLFN model with T hidden nodes. Standard statistical inference cannot be applied here (Section 3.3.2), and we adopted a geometrically interpretable method using the singular value decomposition (SVD).

The use of SVD in NN have been briefly highlighted by Hayashi (1993), Tamura et al. (1993), Psychogios and Ungar (1994), and Teoh et al. (2006). We focus on the use of the SVD as a robust model selection tool in determining if there is hidden node redundancy in the given network. The basic idea is as follows. Suppose that a T -hidden neuron SLFN is fitted based on a sample of size N , $\{(\mathbf{u}_i, y_i), i = 1, 2, \dots, N\}$. As pointed out in Section 3.1, the hidden layer outputs H is a $N \times (T + 1)$ matrix with the t th ($t = 2, 3, \dots, T + 1$) column corresponding to the outcomes from the $(t - 1)$ th neuron; the 1st column of H is a constant vector. Every hidden neuron constructs a hyperplane, and the response vector $(y_1, y_2, \dots, y_N)'$ is a linear combination of the columns of H . Clearly, and in a geometrical sense, the rank of H implies the number of separating hyperplanes of the network. If the rank of H , say r , is less than $T + 1$, it means that redundant hidden neurons have been included in the model and that we can remove $(T + 1 - r)$ neurons from the network without affecting the SSE in (6); Otherwise, each neuron is considered as of value in explaining the target response.

However, strict definition of rank has little meaning for the numeric matrix H subject to estimation errors, and what we would like to estimate is the so-called effective rank of H , which can be obtained by SVD. Specifically, applying SVD onto H gives $H = G \times S \times V$ where G (an $N \times N$ matrix) is known as the left singular vectors of H , and V (an $N \times N$ matrix) the right singular vectors of H . Both G and V are orthonormal. The matrix S is a diagonal matrix with unique nonnegative entries ordered in decreasing magnitude:

$$S_{N \times (T+1)} = \text{diag}(s_1, s_2, \dots, s_{T+1}) \quad (7)$$

where $s_1 \geq s_2 \geq \dots \geq s_r \geq 0 = s_{r+1} = \dots = s_{T+1}$.

The effective rank of H is determined by observing that the r largest singular values are nonzero, whereas the $T + 1 - r$ smallest singular values are zero. Thus, a threshold has to be set to determine the effective rank r . Since our fitting process insures that for each T -hidden node network there can be at most one redundant neuron, we adopted the following rule in our experiments:

- If $s_{T+1}/s_1 < \epsilon$, then declare $r < T + 1$, and one redundant node should be eliminated from the current model;
- Otherwise, declare $r = T + 1$, and there is no redundancy in the current model.

Here, ϵ is a small positive value chosen by the user. In our experiments, ϵ is set as 3%. Interested readers are referred to Golub and Van Loan (1996), Teoh et al. (2006), and Konstantinides and Yao (1988) for detailed discussions of the SVD of a numeric matrix, and the SVD-based decision rules to determine the model complexity.

4. Procedure for estimating the CT-x-PM surface

For convenience of discussion, we rewrite the SLFN model (4) introduced in Section 3.1:

$$Y(\mathbf{u}) = f(\mathbf{u}, \theta) + \varepsilon = \sum_{t=0}^T b_t \cdot h(\mathbf{w}_t' \mathbf{u}) + \varepsilon, \quad \mathbf{u} \in \Gamma_v. \quad (8)$$

Recall that $\mathbf{u} = (x, \alpha)$ is considered as the controllable vector with the feasible region being $\Gamma_v = [x_L, x_U] \times \Omega_v$. The stochastic response $Y(\mathbf{u})$ represents the CT estimate obtained from running simulation under \mathbf{u} , and $Y(\mathbf{u})$ is normally distributed with variance σ^2 .

We devise a sequential procedure to metamodel the CT-x-PM surface within Γ_v using the SLFN (8). It is assumed that the experimentation is driven by limited computation budget, that is, the procedure will be terminated once the user runs out of computation time.

4.1. Multistage procedure

The inputs/outputs of the procedure are given as follows.

- Inputs:** Simulation model of the manufacturing system being investigated; the product type (assumed to be type 1) of particular interest to the user; operational region Γ_v ; desired constant variance σ^2 for $Y(\mathbf{u})$ with $\mathbf{u} \in \Gamma_v$.
- Outputs:** A set of CT-x-PM surfaces $\{\hat{c}_k(x, \alpha), k = 1, 2, \dots, K\}$ with $x \in [x_L, x_U]$ and $\alpha \in \Omega_v$.

Assuming that $E[Y(\mathbf{u})] = c_1(x, \alpha)$, $\mathbf{u} \in \Gamma_v$, is the response surface of primary interest, we next describe the procedure centering on the estimation of $c_1(x, \alpha)$. The CT-x-PM surfaces for products of other types are a desirable by-product of the same simulation procedure given below.

- Step 1.** We start with a small pilot design with, say N_0 , design points $\mathcal{A}_0 = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N_0}\}$ within the operational region Γ_v . The selection of the initial design will be discussed in Section 4.2. Set $\mathcal{A} = \mathcal{A}_0$ and $N = N_0$.
- Step 2.** For each design point $\mathbf{u} \in \mathcal{A}$, we obtain a CT estimate $Y(\mathbf{u})$ for product 1, which is the stochastic output for model (8). The two-stage experimentation described below is used to obtain $Y(\mathbf{u})$ with prespecified variance σ^2 .

At the first stage, we generate, say n_0 i.i.d (independently and identically distributed) sample data $\{\overline{CT}_1(\mathbf{u}), \overline{CT}_2(\mathbf{u}), \dots, \overline{CT}_{n_0}(\mathbf{u})\}$ with $\overline{CT}_j(\mathbf{u})$ being the average cycle time of product 1 estimated from a single simulation run:

$$\overline{CT}_j(\mathbf{u}) = Q(\mathbf{u})^{-1} \sum_{q=1}^{Q(\mathbf{u})} CT_{jq}(\mathbf{u}). \quad (9)$$

Here $CT_{jq}(\mathbf{u})$ represents the individual cycle time of the q th job collected in the steady-state of the j th simulation replication; the distribution of the steady-state CT depends on \mathbf{u} only. For each simulation run, cycle time observations during the initial transient state were discarded, and the cycle times of $Q(\mathbf{u})$ jobs simulated in steady-state were collected. See Law and Kelton (2000) for the

methods to determine the length of initial state and the value of $Q(\mathbf{u})$ in steady-state. We assume that for a given \mathbf{u} the individual cycle times $CT_{jq}(\mathbf{u})$ are identically distributed, although not in general independent within a replication. In our experiments, a different random stream was assigned to each simulation run to insure independence across replications, and we have

$$Y_0(\mathbf{u}) = \frac{1}{n_0} \sum_{j=1}^{n_0} \overline{CT}_j(\mathbf{u}). \quad (10)$$

Denote the sample variance for $\{\overline{CT}_1(\mathbf{u}), \overline{CT}_2(\mathbf{u}), \dots, \overline{CT}_{n_0}(\mathbf{u})\}$ as $\hat{\sigma}_0^2$, then $\text{Var}[Y_0(\mathbf{u})] = \hat{\sigma}_0^2/n_0$. The sample size $n(\mathbf{u})$ that is likely to provide a desired variance σ^2 for $Y(\mathbf{u})$ is estimated as $n(\mathbf{u}) = \lceil \hat{\sigma}_0^2/\sigma^2 \rceil$.

At the second stage, perform $n(\mathbf{u}) - n_0$ simulation replications at \mathbf{u} , and recalculate

$$Y(\mathbf{u}) = \frac{1}{n(\mathbf{u})} \sum_{j=1}^{n(\mathbf{u})} \overline{CT}_j(\mathbf{u}). \quad (11)$$

The resulting CT estimate $Y(\mathbf{u})$ is used as stochastic output in model (8) which is approximately normally distributed with variance σ^2 . The normality of $Y(\mathbf{u})$ can be justified by appealing to the Central Limit Theorem for identically and independently distributed random variables.

The desired constant variance σ^2 is a user-specified parameter, and we recommend setting σ^2 in such a way that a high relative precision, say $\gamma\%$, is obtained for $Y(\mathbf{u})$. Suppose that $E[Y(\mathbf{u})] = c_1(x, \alpha)$, $\mathbf{u} \in \Gamma_v$ falls into the range of $[c_{\min}, c_{\max}]$, which can be roughly estimated from the user's past experience. Then σ^2 can be set as:

$$\frac{2\sigma}{c_{\min}} = \gamma\% \quad (12)$$

In our experiments, we set $\gamma\% = 4\%$, and used $\sigma = c_{\min} \times 2\%$. Hence, the $Y(\mathbf{u})$ obtained at each design point is a highly precise CT estimate.

Step 3. Based on the N -point sample collected so far $\{Y(\mathbf{u}_1), Y(\mathbf{u}_2), \dots, Y(\mathbf{u}_N)\}$, we estimate the SLFN (8) for the target surface $c_1(x, \alpha)$ following the iterative model-fitting strategy outlined in Fig. 4.

Step 4. If the computing budget has been exhausted, stop; otherwise, we expand the current N -point design by including one more design point $\mathbf{u}_{N+1} \in \Gamma_v$. The additional design point \mathbf{u}_{N+1} is selected following the method described in Section 4.3. For \mathbf{u}_{N+1} , repeat the two-stage experiments described in Step 2 to obtain the CT estimate $Y(\mathbf{u}_{N+1})$. Set $N = N + 1$, and then go back to Step 3.

In the procedure described above, we focus on the estimation of $E[Y(\mathbf{u})] = c_1(x, \alpha)$, and collect the CT estimates $\{Y(\mathbf{u}); \mathbf{u} \in \mathcal{A}\}$ for product 1 from simulation. When we investigate multi-product systems, such CT estimates can be obtained simultaneously for each type of products, except that the variance of the CT estimates will not be as well controlled for products of other types since the simulation experiments in Step (2) are driven by the prespecified variance σ^2 for product 1. Nevertheless, a different SLFN can be fitted for a different type of products, and the procedure given above is able to provide the CT-x-PM surfaces for all types of products, $\{\hat{c}_k(x, \alpha), k = 1, 2, \dots, K\}$ with $x \in [x_L, x_U]$ and $\alpha \in \Omega_v$.

4.2. Initial design

The operational region Γ_v is a linear simplex defined by the utilization range $[x_L, x_U]$ and the PM subregion Ω_v (Section 3.1). For

such a simplex, we follow the recommendation of Myers and Montgomery (2002), and select the initial design points from a candidate set, say $\mathcal{C} = \{\mathbf{u}_1^*, \mathbf{u}_2^*, \dots, \mathbf{u}_N^*\}$, which provides a good coverage of the input space. In our experiments, \mathcal{C} includes the following points of the simplex Γ_v : extreme vertices, edge centers, constraint plane centroids, overall centroid and axial points.

Given the constraints (24) that define the simplex, we can use the CONVRT and CONAEV algorithms developed by Piepel (1988) to find the vertices, edge centers, and all other centroids of the simplex. In our procedure, the initial design points will be selected as a subset of these candidate points in \mathcal{C} . Let \mathcal{A}_0 denote the set of initial design points of size N_0 within the simplex. We propose some additional considerations on the initial set of design points:

- To avoid extrapolation, \mathcal{A}_0 must include all the N_v extreme vertices.
- The number of initial design points N_0 should be sufficiently large to allow for the fitting of a SLFN with two hidden neurons.
- Apparently, the number of design points required for an adequate surface fitting depends on the span of input region Γ_v . For instance, with the same PM subregion Ω_v , the number of design points required in the case with $[x_L, x_U] = [0.75, 0.85]$ is likely to be much less than that in the case with $[x_L, x_U] = [0.75, 0.95]$. This is due to the fact that the CT tend to increase dramatically within the utilization range of $[0.9, 0.95]$ (Yang et al., 2007). Hence, the user can utilize her knowledge of the response surface and adjust the initial sample size accordingly.

Thus, we require $N_0 \geq \max\{1 + 2 \times (K + 2), N_v\}$. The additional $N_0 - N_v$ non-vertex points are selected from \mathcal{C} using a *maxmin* criterion which maximizes the minimum distance between any two points.

4.3. Design augmentation

We initiate the experiments with a pilot design (Section 4.2), and then augment the design by sequentially including one more point at a time until the computing budget is exhausted.

To the best of our knowledge, the only work that addresses the sequential design issues for NN fitting is Witczak (2006), which proposed using the sequential D-optimum design (Seber and Wild, 2003) as if NN is no different from a traditional nonlinear regression model. Denote D as the matrix of partial derivatives of the SLFN with element D_{ij} given as

$$D_{ij} = \frac{\partial f(\mathbf{u}_i, \hat{\theta})}{\partial \theta_j} \quad (13)$$

where $f(\mathbf{u}_i, \hat{\theta})$ is defined in model (8), \mathbf{u}_i ($i = 1, 2, \dots, N$) is the i th design point in the input region Γ_v , and θ_j denotes the j th NN model parameter. D-optimality is based on the premise that the variance of the least-square parameter estimators $\hat{\theta}$ can be approximated using the standard statistical inference method, i.e., $\hat{\theta}$ is approximately normally distributed with $\text{Var}[\hat{\theta}] = \sigma^2(D'D)^{-1}$. The D-optimality criterion is to maximize $|D'D|$, which is inversely proportional to the size of a confidence ellipsoid for the least-squares estimates $\hat{\theta}$. Unfortunately, as pointed out in Section 3.3.2, the theoretical basis for D-optimality is very much questionable for NN models: with the lack of local identification, the distribution of the estimated NN parameters $\hat{\theta}$ cannot be approximated by multi-normality and standard methods cannot be used to derive the statistical inference on $\hat{\theta}$. Aside from statistical invalidity, the D-optimality design will almost inevitably encounter enormous computational difficulties since the matrix D is likely to be ill-conditioned due to the interdependency

among network parameters. Therefore, a conventional D-optimal design is not likely to be successful when applied to NN modeling. In the remainder of this section, we present a simple strategy to design simulation experiments sequentially for the fitting of the SLFN models, the efficiency of which will be demonstrated in Section 5.

Suppose that the current design consists of N points and the simulation data obtained so far are given as $\{(\mathbf{u}_i, Y_i); i = 1, 2, \dots, N\}$. While more computation time is available, we will expand the current design by running simulation experiments at an additional input setting \mathbf{u}_{N+1} . The best choice of \mathbf{u}_{N+1} depends on the true target surface which unfortunately is unknown. In our design augmentation, we will utilize the information obtained from the available data set (i.e., the best estimated response surface fitted from the current data) to guide the choice of \mathbf{u}_{N+1} .

Suppose that the best SLFN model fitted from the current data set is $f(\mathbf{u}_i, \hat{\theta})$ which consists of T hidden nodes. The essential ingredient of our sequential experiment design is the hidden layer output matrix \hat{H} , which is an $N \times (T + 1)$ matrix (Section 3.1) with the i th row being

$$\hat{\eta}(\mathbf{u}_i) = [h(\hat{\mathbf{w}}'_0 \mathbf{u}_i), h(\hat{\mathbf{w}}'_1 \mathbf{u}_i), \dots, h(\hat{\mathbf{w}}'_T \mathbf{u}_i)] \quad (14)$$

Recall that $h(\mathbf{w}'_0 \mathbf{u}) = 1$ for any \mathbf{u} . The SLFN model (4) can be written in terms of H as

$$\mathbf{Y} = \hat{H}\mathbf{c} + \boldsymbol{\varepsilon} \quad (15)$$

where \mathbf{Y} is the $N \times 1$ output vector, and \mathbf{c} is the $(T + 1) \times 1$ parameter vector as defined in model (4). In this sense, a linear regression is performed: projecting \mathbf{Y} onto the subspace spanned by the columns of \hat{H} .

Including one more design point \mathbf{u}_{N+1} will add an additional row $\hat{\eta}(\mathbf{u}_{N+1})$ to the \hat{H} matrix, and $\hat{\eta}(\mathbf{u}_{N+1})$ can be approximately evaluated using the fitted NN model $f(\mathbf{u}_i, \hat{\theta})$. The experiment design question we intend to answer is: How to determine \mathbf{u}_{N+1} so that the resulting expanded H matrix will have the most desirable property in terms of modeling the output as a linear combination of the H columns? We used the D-optimality criterion to measure the goodness of the expanded H matrix, and \mathbf{u}_{N+1} is determined by

$$\max_{\mathbf{u}_{N+1}} \left| \begin{pmatrix} \hat{H} \\ \hat{\eta}(\mathbf{u}_{N+1}) \end{pmatrix} \begin{pmatrix} \hat{H} \\ \hat{\eta}(\mathbf{u}_{N+1}) \end{pmatrix}' \right| \quad (16)$$

Given the matrix \hat{H} , (16) is equivalent to

$$\max_{\mathbf{u}_{N+1}} \hat{\eta}(\mathbf{u}_{N+1})(\hat{H}'\hat{H})^{-1}\hat{\eta}(\mathbf{u}_{N+1})'. \quad (17)$$

Note that the fitting process described in Fig. 4 insures that the matrix H has full column rank, and hence $H'H$ is invertible. The solution of (17) can be approximated by evaluating the objective function over a fine grid of $\mathbf{u} \in \Gamma_v$.

5. Empirical evaluation

Here, through empirical studies, we demonstrate the effectiveness of the proposed NN metamodeling, and compare the NN-based approach in this work with the traditional regression-based metamodeling developed in our previous work (Yang et al., accepted for publication).

Two different systems, an analytically tractable Jackson network and a real fab model, are explored. In our experiments, it is assumed that the product-mix is not subject to additional linear constraints (23) imposed by practical considerations of production planning. Hence, the feasible input region being considered here is larger than otherwise, and we are resolving a more challenging problem from the perspective of response surface modeling.

5.1. Jackson network models

We consider a 4-product and 3-station Jackson network, for which the true CT- x -PM surface is known from queueing theory and hence provides a benchmark to evaluate the resulting SLFN metamodels. The configuration of this Jackson queueing model is specified in Appendix A.5.

First, queueing analysis is performed to partition the PM region into constant-BN subregions. Each station can serve as a BN and the PM region is divided into 3 subregions with Ω_v being dominated by BN station v ($v = 1, 2, 3$). Hence, we have three constant-BN input subregions $\Gamma_v = [x_L, x_U] \times \Omega_v$ ($v = 1, 2, 3$). In addition, the system capacity $u^*(\boldsymbol{\alpha})$ is derived from this analytical analysis so that conversion between the utilization x and throughput is possible (Section 2.2).

Second, we apply the metamodeling procedure on each subregion Γ_v and estimate smooth CT- x -PM surfaces $c_k(x, \boldsymbol{\alpha})$ ($k = 1, 2, 3, 4$) within Γ_v ($v = 1, 2, 3$). The user-specified parameters for the procedure (Section 4) are given as follows:

- Products of type 1 are considered as of particular interest to the user.
- The desired constant variance σ^2 is set following Eq. (12).
- Two different utilization ranges, $[x_L, x_U] = [0.75, 0.85]$ and $[x_L, x_U] = [0.75, 0.95]$ are considered in our experiments. The former is the typical range within which semiconductor manufacturers run their facility (Hopp, 2007). The latter leads to a much more difficult response surface due to the extreme edge steepness of the CT surface at high system utilization (Section 4.2). We next present these two cases, respectively.

Due to space constraints, here we present the estimation results for Γ_1 . Similar results have been obtained for Γ_2 and Γ_3 .

Case 1: $[x_L, x_U] = [0.75, 0.85]$

The initial design set \mathcal{A}_0 in Γ_1 is determined as described in Section 4.2. The initial number of design points is selected as 16, which is the number of vertices for simplex Γ_1 and which also allows for the fitting of a SLFN with two hidden neurons. For each $\mathbf{u} \in \mathcal{A}_0$, simulation is performed to obtain $Y(\mathbf{u})$, the CT estimate for type 1 products, with prespecified variance σ^2 (Section 4.1, Step 2). Additional design points are added one at a time (Section 4.3) until the computation time has been exhausted.

To illustrate the accuracy of the estimated SLFNs and the efficiency of our experiment design strategy, we display in Fig. 5 the evolution of the model estimation errors as more points are incorporated into the design. Specifically, about 54,000 check points evenly distributed within Γ_1 are used to evaluate the resulting SLFNs fitted from, say N design points. At each check point, the relative error, which is defined as

$$re = \frac{\hat{c}_1(x, \boldsymbol{\alpha}) - c_1(x, \boldsymbol{\alpha})}{c_1(x, \boldsymbol{\alpha})}, \quad (18)$$

is calculated. Here, $c_1(x, \boldsymbol{\alpha})$ is the true CT from queueing theory, and $\hat{c}_1(x, \boldsymbol{\alpha})$ is the CT estimated from the fitted SLFN. Fig. 5 gives histograms of the relative errors over the 54,000 check points with graphs a, b, c, and d corresponding to $N = 16, 17, 19$, and 21 design points. For instance, Fig. 5a plots the histogram of errors for the model fitted from the initial design consisting of 16 points. Evidently, the accuracy of the metamodels is improved in an efficient manner as more design points are incorporated. The size of the maximum deviation starts from 0.15 in Fig. 5a, and decreases to 0.04 in Fig. 5d.

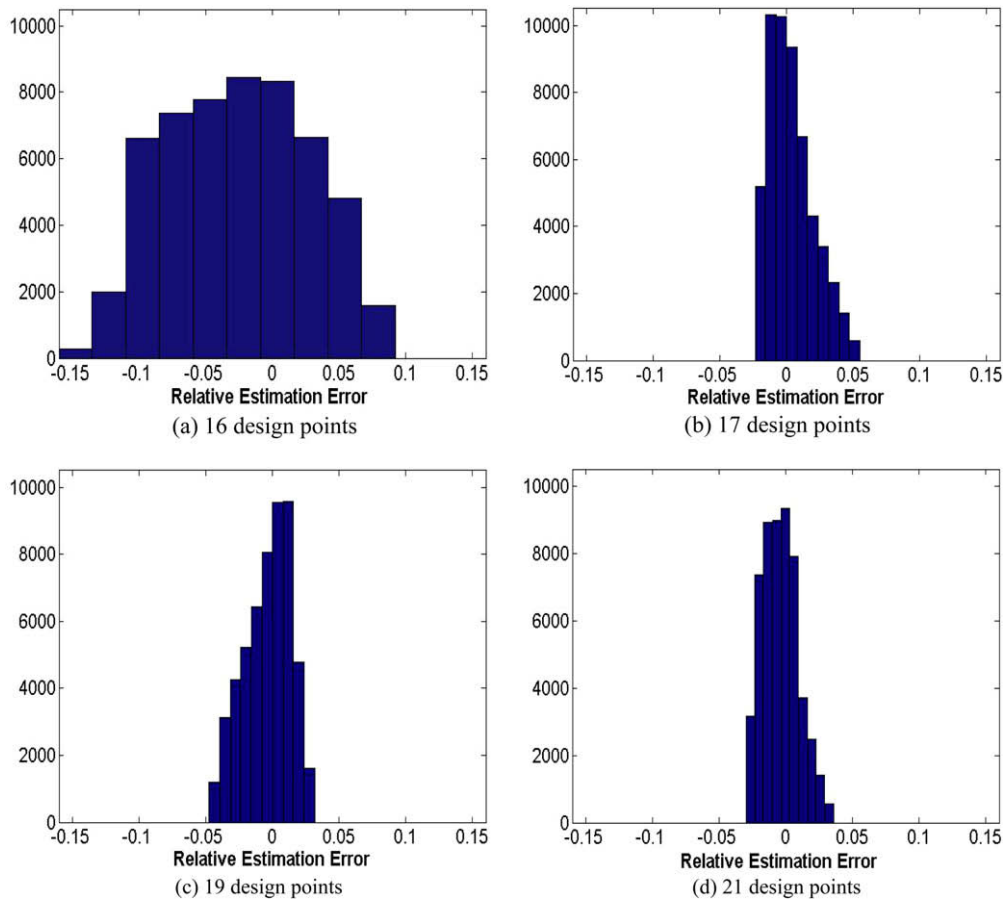


Fig. 5. Histograms of the relative model estimation errors (Case 1).

The estimated SLFNs here include two hidden neurons, and we present the model fitted from the 21 design points for an example (corresponding to Fig. 5d):

$$\hat{c}_1(x, \alpha) = 3.80 + \frac{60.38}{1 + \exp(-6.49 + 8.16x + 8.69\alpha_1 - 8.65\alpha_2 - 5.04\alpha_3)} + \frac{70.25}{1 + \exp(-8.80 + 8.98x + 2.21\alpha_1 - 2.20\alpha_2 - 2.71\alpha_3)}. \quad (19)$$

The simulation procedure, though driven by the precision of CT for type 1 products, also provides the data for fitting the CT-x-PM surfaces of all types of products $\{\hat{c}_k(x, \alpha), k = 1, 2, \dots, K\}$. In our experiments, all these estimated response surfaces are highly accurate. Compared to this NN metamodeling approach, the regression-based methods developed in Yang et al. (accepted for publication) requires about three times as much simulation effort to achieve the fitted surfaces of about the same quality.

Case 2: $[x_L, x_U] = [0.75, 0.95]$

We extend the upper bound of the utilization range to 0.95. Extending the lower bound will not introduce any additional difficulties in the CT-x-PM modeling and will not be further discussed here. But since CT tends to explode when system utilization is pushed up to the limit (Hopp and Spearman, 2008), the CT-x-PM surfaces covering such a high utilization range are difficult to model. In our experience, the traditional regression metamodels in Yang et al. (accepted for publication) is not able to generate a good fit for the target surfaces over a utilization range as wide as $[0.75, 0.95]$.

Due to the large operational region for this case, an initial design of relatively large size is adopted (Section 4.2): 37 design points, which allows for the fitting of a SLFN model with 6 hidden neurons. The design is then expanded by including one additional point at a time. As in Case 1, we provide the histograms of the estimation errors in Fig. 6 where about 112,000 check points are used. Evidently, the SLFN is better estimated as more design points are incorporated, and the relative error drops to within 0.05 once a total of 53 design points are collected. The resulting SLFN obtained from 53 design points has six hidden neurons.

5.2. A semiconductor manufacturing system

We consider a semiconductor wafer fab simulation (available at www.eas.asu.edu/masmlab/) and characterize the system's performance by its CT-TH surfaces. Three types of product families go through this system, and our objective is to generate $\{c_k(x, \alpha), k = 1, 2, 3\}$. We chose to present a three-product case out of two considerations: (i) At the aggregate level of production planning, products are usually grouped into a relatively small number of families; (ii) With three different product types, the feasible product mix region can be clearly and graphically illustrated, as will be seen next.

First, the analytical engine provided by factory explorer (an integrated capacity, cost and discrete-event simulation software package particularly suitable for modeling wafer fabs; for details, please refer to <http://www.wwk.com/productfx.html>) is used to perform the capacity/bottleneck analysis of the fab model. As shown in Fig. 7, the PM region is divided into 4 constant-BN subregions with each one defined by a number of linear constraints. For

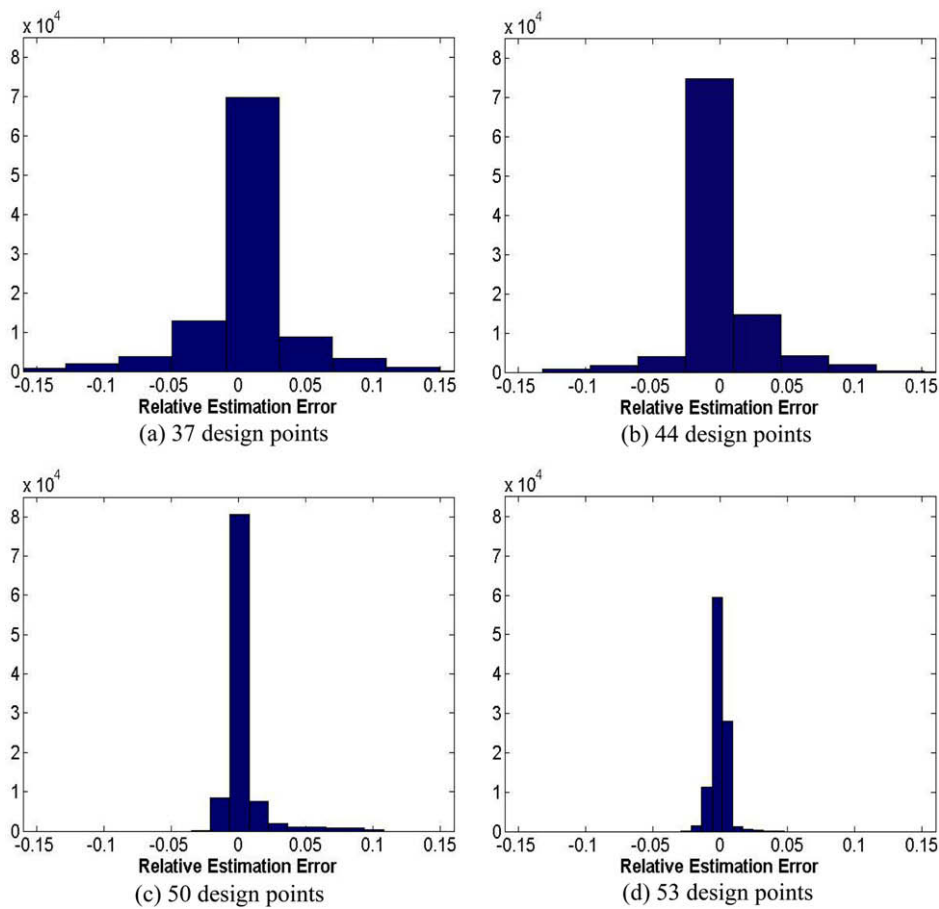


Fig. 6. Histograms of the relative model estimation errors (Case 2).

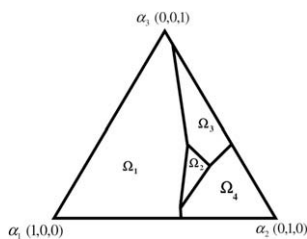


Fig. 7. Division of PM region for the wafer fab model.

each PM vector α , factory explorer also analytically estimates the system capacity $u^*(\alpha)$, which allows for the conversion between the throughput λ and system utilization x (Section 2.2).

The input parameters to the procedure are set up as follows. Suppose that product 1 is of primary interest. The utilization varies within $[x_L, x_U] = [0.75, 0.85]$, which is a typical range under which a real fab system is running (Hopp, 2007). Since the CT for this fab roughly ranges from 300 to 450 h. Following the rule in (12), σ is set at $6 \approx c_{min} \times \gamma\% / 2$ h with $c_{min} = 300$ and $\gamma\% = 4\%$.

We present here the modeling results for $\Gamma_1 = [x_L, x_U] \times \Omega_1$. Since the true CT–TH surfaces is unknown in this case, grid points evenly distributed over Γ_1 are selected to check the lack of fit of the estimated surfaces at those locations. At each check point, substantial simulation effort, completely independent of that required by the metamodeling procedure, was invested to obtain CT estimate whose standard error was essentially zero. These CT estimates

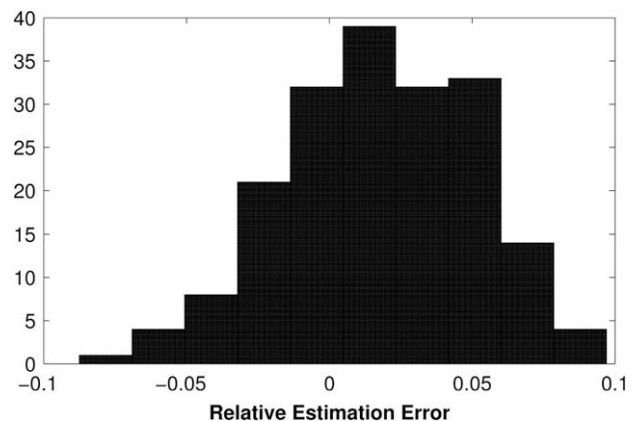


Fig. 8. Histogram of the relative model estimation errors for the semiconductor fab system.

are considered as “nearly true”, and used to evaluate the CT estimates obtained from the SLFN metamodels $\hat{c}_1(x, \alpha)$, $\alpha \in \Omega_1$. Fig. 8 shows the histogram of the relative deviations of the estimated cycle times from their “true” values. Among the 180 check points, all the relative deviations fall within the range of $[-10\%, 10\%]$ with most of them within $[-5\%, 5\%]$.

In our procedure, simulation was performed at 11 design points in Γ_1 for data collection, and the resulting SLFN includes two hidden neurons. We emphasize that once the metamodeling procedure is complete, no more simulation is needed for the decision

making process. The resulting metamodel $\hat{c}_k(x, \alpha)$ is a mathematical equation that provides a direct functional relationship between the CT and TH. In this case, about 5-h simulation time (on a computer with processor speed of 3 GHz) was invested in those 11 design points for the generation of the CT–TH surfaces $\{\hat{c}_k(x, \alpha), k = 1, 2, 3\}$ within Γ_1 . The simulation time will be substantially driven down by either considering a more restricted region than Γ_1 (Fig. 7), which is likely to be the case in practice, or prespecifying a slightly larger value for σ , the desired error of the CT estimate at each design point.

We regard this as very good results considering the following. Only 11 design points in Γ_1 were utilized to estimate the SLFN $\hat{c}_1(x, \alpha)$, and a high accuracy as demonstrated in Fig. 8 has been obtained. To provide a baseline as to how many design points may be required by other types of metamodels, consider a full quadratic model with two-factor interactions: With three independent variables x , α_1 , and α_2 , it requires at least 10 distinct design points (equal to the number of unknown parameters in the quadratic model), and in our experience, is far from giving an adequate approximation of the target surface. The traditional nonlinear regression-based method in Yang et al. (accepted for publication) requires more than 30 design points to achieve the response surface of about the same quality for this case.

5.3. The selection of model families

The objective of metamodeling is to provide a sufficiently good functional approximation (metamodel) for the target response surface using the smallest amount of simulation effort. There are different types of metamodels ranging from the linear regression to those powerful modeling tools such as Kriging (Cressie, 1993) and NN models. How do we select a best model family to approximate the response surface of interest? In our CT–TH modeling, we explored the use of conventional (Yang et al., accepted for publication) as well as non-conventional models, and we hope that our experience could give some insights to this question.

First, it is important to examine the geometry of the target surface and evaluate the potential of the selected model to approximate such a surface. Initially, recognizing the complexity of the CT–TH surface over PM region, we used Kriging hoping that this powerful model is able to adapt well to the entire surface. After many unsuccessful experiments, we concluded that Kriging cannot provide an adequate fit unless using a highly dense sample, which is computationally impractical. This motivates us to divide the PM region into a number of subregions and examine them separately. Within each subregion, the target surface is differentiable and bowl-shaped, which fosters some specific functional forms of the metamodels. Both the nonlinear regression in Yang et al. (accepted for publication) and the SLFNs in this work are adopted due to their potential ability to capture bowl shapes.

The comparison between the regression and NN models brings up the second consideration in selecting a model family: robustness, which we interpret as:

1. The ability to approximate the target surface over a wide range of inputs.
2. The ability to provide an adequate fit with minimum design points (simulation effort); the goodness of the fit is not sensitive to the locations of the design points.

In our experience (Sections 5.1 and 5.2), NN performs better than the nonlinear regression model in terms of robustness, which is somewhat surprising to us. Take the Jackson networks as an example. In Yang et al. (accepted for publication), the regression model takes the form of a sum of ratio functions, which is adapted

directly from the true CT–TH relationships given in formula (3), and hence can be considered as more “similar” to the target function than the NN, at least in the case of Jackson networks. Nevertheless, as illustrated in Section 5.1, the robustness of NN is far better than that of the nonlinear regression. To find the reason of these somehow counterintuitive results, it may require a thorough study of the model identification issues and a microscopic examination of the least-square fitting processes for both types of models, which is beyond the scope of this paper. Here, we intend to provide some empirical insights to the selection of model families as we did above.

6. Discussions

In this paper, a NN-based metamodeling procedure was developed for the efficient generation of CT–TH profiles for manufacturing systems. Such profiles provide a comprehensive performance evaluation of a given system, and hence support the long-term decisions such as capacity expansion in manufacturing. The metamodeling approach aims at overcoming the shortcomings of the conventional methods used in such decision making contexts, i.e., the lack of fidelity of queueing models to real systems and the heavy computational burden of simulation. The metamodels resulting from our proposed procedure are mathematical equations quantifying the relationship between CT and TH while embodying the high fidelity of simulation.

As a sequent study following our previous work on characterizing manufacturing systems by their CT–TH profiles, this paper explored NN as a metamodeling tool and developed a statistical procedure which is demonstrated to be efficient via numeric experiments. Next, we summarize pros and cons of the NN-based methods, compared the traditional regression-based metamodeling in Yang et al. (accepted for publication). (i) Compared to Yang et al. (accepted for publication), the procedure here is much simpler and thus easier to implement in practice. (ii) As pointed out in Section 5.3, the robustness of NN metamodeling is far better than the regression-based approach. (iii) The main drawback of our NN modeling is that no reliable error variance can be provided for the fitted NN models due to the lack of model identification. As emphasized earlier, we rely on the special geometry of NN to secure well-estimated CT–TH surfaces; and to efficiently obtain these surfaces, we developed a sequential experiment design strategy and a progressive NN fitting process which both have a clear geometrical interpretation. These design and model fitting methods are interesting additions to the literature of NN metamodeling where NN has largely been treated as a black box.

In this paper, attention is centered on quantifying the relationship between the first-moment of cycle time and the throughput. The proposed method can be straightforwardly adapted to estimate the functional dependence of the higher moments of cycle time upon the throughput. Queueing analysis shows that higher moment CT–TH surfaces is also bowl-shaped except that the sides of the bowl is even steeper compared to the first-moment surface. Similar to our experience with Case 2 in Section 5.1, the approach in Yang et al. (accepted for publication) has difficulty approximating a steep higher moment CT–TH surface, whereas the NN methods is able to handle such a target surface.

Acknowledgments

The author gratefully thanks Professor Barry Nelson from Northwestern University for valuable discussion, and Jingang liu from West Virginia University who did part of the computer programming for the empirical studies. The author also thanks

the associate editor and referees for help in clarifying the presentation.

Appendix A

The main contents of this Appendix have been given in Yang et al. (accepted for publication) and are appended here for the readers' convenience.

A.1. Analytical formulation of the manufacturing system

To perform the preliminary queueing analysis (Section 2.1), we treat the manufacturing system as a multi-product queueing network. Suppose that the system (e.g., wafer fab) consists of M stations, and it is designed to process K types of products with each one following a different routing. Despite the complexities involved (e.g., reworks, machine failures, batch processes, etc.), the system can be closely approximated by the following description.

- $\{s_j, j = 1, 2, \dots, M\}$: the number of parallel resources at station j .
- $\{u_{kj}, k = 1, 2, \dots, K; j = 1, 2, \dots, M\}$: the effective service rate of each resource at station j for products of type k .
- $\{\delta_{kj}, k = 1, 2, \dots, K; j = 1, 2, \dots, M\}$: the expected number of visits by type k products to station j .

Using the notation given in Section 2.1, capacity/bottleneck analysis can be performed as follows. We can easily calculate ρ_j , the utilization of station j ($j = 1, 2, \dots, M$). Let $\rho_{kj} = \delta_{kj} / (s_j u_{kj})$, then $\rho_j = \lambda \sum_{k=1}^K \alpha_k \rho_{kj}$. The maximum utilization $\rho_{max} = \max_j \rho_j$ is called the system utilization and is denoted by x in this paper. A station, say station j_{BN} , that reaches ρ_{max} is called a bottleneck (BN) station, that is,

$$j_{BN} = \operatorname{argmax}_j \rho_j = \operatorname{argmax}_j \sum_{k=1}^K \alpha_k \rho_{kj} \quad (20)$$

The stability constraint on the system requires $x = \lambda \sum_{k=1}^K \alpha_k \rho_{kj_{BN}} < 1$, or equivalently,

$$\lambda < 1 / \sum_{k=1}^K \alpha_k \rho_{kj_{BN}} = u^*(\alpha) \quad (21)$$

where $u^*(\alpha)$ is the system capacity, the upper limit on λ (or overall throughput) for stability. Obviously, both capacity $u^*(\alpha)$ and the BN station j_{BN} depend on the system parameters as well as α .

For real manufacturing systems (involving batching, re-entrant flows, machine setups, etc.), existing queueing models can provide an accurate capacity/bottleneck analysis: Given system configuration and PM α , system capacity $u^*(\alpha)$ can be approximated and BN station(s) can be identified. Examples of such queueing are given in Hopp et al. (2002), Kumar and Kumar (2001) and Meng and Heragu (2004).

A.2. The feasible input space

Since the range of interest for utilization x can be easily specified as $[x_L, x_U]$ ($0 < x_L < x_U < 1$), we focus on the feasible region of PM space.

Obviously, product-mix α has to satisfy:

$$\sum_{k=1}^K \alpha_k = 1, \quad \alpha_k \in [0, 1]. \quad (22)$$

Fig. 9(a) illustrates the feasible product-mix region in a 3-product case defined by constraint (22). In practice, the PM is usually subject to additional linear constraints imposed by realistic situations (e.g., lower bounds on release rates). We use the following notation to represent the linear constraints on product-mix

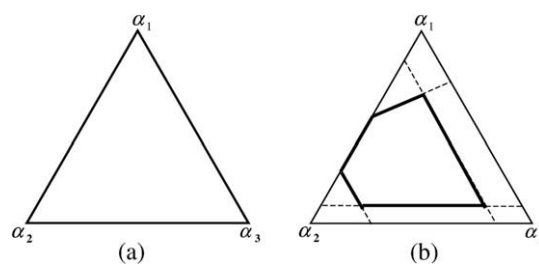


Fig. 9. Feasible product-mix space: unconstrained (a) and constrained (b).

$$A\alpha \leq b \quad (23)$$

where A is a matrix with K columns with each row representing a constraint. Fig. 9(b) gives an example of the more restricted product-mix region defined by (22) and (23).

A.3. Partitioning the product-mix (PM) space

Production systems are usually constrained by one or more bottleneck resources. A bottleneck (BN) is usually a facility or resource which most constrains the production flow, and it plays a key role in determining the overall performance of the manufacturing system. As we change the PM, the BN may shift from one resource to another, which complicates the way that PM affects the cycle time. As explained in Section 2.3, within an PM region where no BN shift occurs, $c_k(x, \alpha)$ tends to be smooth and differentiable with respect to x as well as α . For the purpose of modeling the CT- x -PM surface, we divide the PM space into a number of subregions with each one dominated by a different BN station or stations, and fit the response surface for each subregion individually.

Suppose the PM region of feasibility is defined as

$$\Omega = \{\alpha | \alpha \text{ satisfies constraints (22) and (23)}\}.$$

Following the definition of BN station provided by (20), the subregion

$$\Omega_v = \{\alpha | \alpha \in \Omega \text{ and } \alpha \text{ mix makes station } v \text{ a BN}\}$$

is given as the collection of α that satisfies

$$\sum_{k=1}^K \alpha_k = 1 \quad (24)$$

$$A\alpha \leq b$$

$$\rho_v \geq \rho_j \quad j = 1, 2, \dots, M \text{ and } j \neq v.$$

Following up on the 3-product example discussed in Section A.2, we further suppose that the system consists of 3 stations. It can be shown that for such a system the feasible region displayed in Fig. 9b could be divided in three different ways as shown in Fig. 10 depending on the system parameters.

For real systems, as established in Appendix A.1, existing queueing models can be used to derive the station utilization ρ_j ($j = 1, 2, \dots, M$) as a function of PM α . Thus, the partition of the PM region into constant-BN subregions can be realized from analytical analysis prior to the simulation metamodeling. For the case study in Section 5.2, we use the analytic engine in factory explorer to divide the PM region for a wafer fab.

A.4. CT- x -PM surface of a Jackson network

Here we graphically illustrate the shape of the CT- x -PM surface through a Jackson network example.

We consider a 3-product and 3-station Jackson network, for which the true CT- x -PM surface is known from queueing theory. The system configuration is specified in Table 1 following the

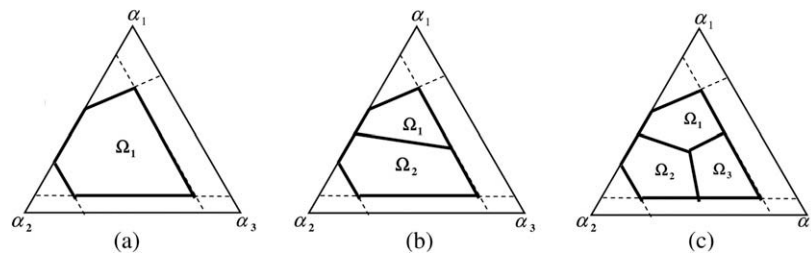


Fig. 10. Division of the feasible product-mix region.

Table 1
Three-station Jackson queueing model.

Station 1	Station 2	Station 3
$s_1 = 1$	$s_2 = 1$	$s_3 = 1$
$u_1 = 4$	$u_2 = 3$	$u_3 = 2.8$
$\delta_{11} = 1$	$\delta_{12} = 2$	$\delta_{13} = 3$
$\delta_{21} = 3$	$\delta_{22} = 2$	$\delta_{23} = 1$
$\delta_{31} = 2$	$\delta_{32} = 1$	$\delta_{33} = 1$

Table 2
Three-station Jackson queueing model.

Station 1	Station 2	Station 3
$s_1 = 1$	$s_2 = 1$	$s_3 = 1$
$u_1 = 4$	$u_2 = 3$	$u_3 = 2.8$
$\delta_{11} = 1$	$\delta_{12} = 2$	$\delta_{13} = 3$
$\delta_{21} = 3$	$\delta_{22} = 2$	$\delta_{23} = 1$
$\delta_{31} = 2$	$\delta_{32} = 1$	$\delta_{33} = 1$
$\delta_{41} = 2$	$\delta_{42} = 2$	$\delta_{43} = 2$

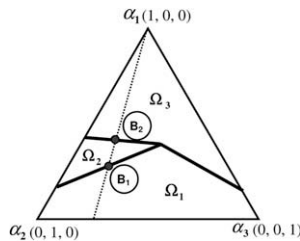


Fig. 11. Division of the product-mix space and a constant-ratio product-mix path.

$\alpha_2 : \alpha_3 = 3 : 1$, and vary α_1 from 0 to 1, we obtain a PM path as the dotted line in Fig. 11. Along this path, we plot $c_k(0.8, \alpha)$ ($k = 1, 2, 3$), the cycle time at throughput $x = 0.8$, against α_1 , and the resulting CT-PM curves are given in Fig. 12. Obviously in Fig. 12, the CT-PM curves are smooth and differentiable except at BN shift points B_1 and B_2 , which are also marked in Fig. 11. We can change the ratio of $\alpha_2 : \alpha_3$, and plot CT-PM curves similar to those obtained in Fig. 12. This graphically demonstrates our conclusion in Section 2.3, which motivates us to model each subregion Ω_v separately.

A.5. A 4-product and 3-station Jackson network

Following the notation defined in Appendix A.1, the system configuration of a 4-product and 3-station Jackson network is specified in Table 2.

References

Anders, U., Korn, O., 1999. Model selection in neural networks. *Neural Networks* 12, 309–323.
 Atherton, R.W., Dayhoff, J.E., 1986. Signature analysis: simulation of inventory cycle, time, and throughput trade-offs in wafer fabrication. *IEEE Transactions On Components, Hybrids, Manufacturing Technology* CHMT-9 (4), 498–507.
 Bitran, G.R., Tirupati, D., 1988. Multiproduct queueing networks with deterministic routing: decomposition approach and the notion of interference. *Management Science* 34 (1), 75–100.
 Chen, H., Harrison, J.M., Mandelbaum, A., Ackere, A.V., Wein, L.M., 1988. Empirical evaluation of a queueing network model for semiconductor wafer fabrication. *Operations Research* 36 (2), 202–215.
 Connors, D., Feigin, G., Yao, D., 1996. A queueing network model for semiconductor manufacturing. *IEEE Transactions On Semiconductor Manufacturing* 9 (3), 421–427.
 Cressie, N., 1993. *Statistics for Spatial Data*. Wiley, New York.
 Curry, B., Morgan, P.H., 2006. Model selection in neural networks: some difficulties. *European Journal of Operations Research* 170 (2), 567–577.
 Davidson, R.J.G., MacKinnon, 1993. *Estimation and Inference in Econometrics*. Oxford University Press.
 Fowler, J., Rose, O., 2004. Grand challenges in modeling and simulation of complex manufacturing systems. *Transactions of the Society for Computer Simulation International* 80 (9), 469–476.
 Golub, G.H., Van Loan, C.F., 1996. *Matrix Computations*. third ed. The Johns Hopkins University Press.
 Hayashi, M., 1993. A fast algorithm for the hidden units in a multilayer perceptron. *Proceedings of 1993 International Joint Conference on Neural Networks* 1, 339–342.
 Henderson, S.G., Nelson, B.L., 2006. *Handbooks in Operations Research and Management Science: Simulation*. Elsevier Science, Amsterdam, Netherlands.
 Hopp, W.J., 2007. *Supply Chain Science*. McGraw-Hill/Irwin, New York.

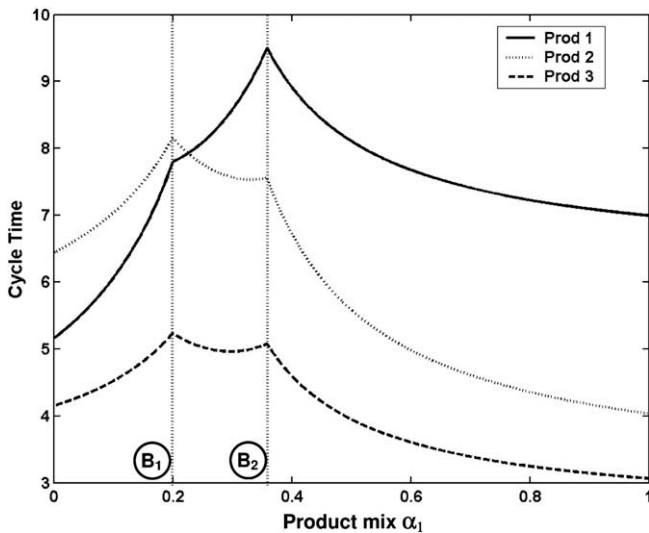


Fig. 12. CT-PM curves.

notation defined in Appendix A.1. From analytical queueing analysis, the PM region can be partitioned into constant-BN subregions. Fig. 11 shows the division of product-mix space for this example. Each station can serve as a BN and the PM region is divided into 3 subregions with Ω_v being dominated by BN station v ($v = 1, 2, 3$).

For an open Jackson network, the CT- x -PM surface is given by (2). Apparently, (2) is always continuous and differentiable with respect to $x \in [x_L, x_U]$, so we focus our attention on the CT-PM surface with a given x . We take $x = 0.8$ for an example. If we fix

- Hopp, W., Spearman, M.L., 2008. *Factory Physics*. third ed. McGraw-Hill/Irwin, New York.
- Hopp, W.J., Spearman, M.L., Chayet, S., Donohue, K.L., Gel, E.S., 2002. Using an optimized queueing network model to support wafer fab design. *IEE Transactions* 34, 119–130.
- Konstantinides, K., Yao, K., 1988. Statistical analysis of effective singular values in matrix rank determination. *IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-36* (5), 757.
- Kumar, S., Kumar, P.R., 2001. Queueing network models in the design and analysis of semiconductor wafer fabs. *IEEE Transactions on Robotics and Automation* 17 (5), 548–561.
- Law, A.M., Kelton, W.D., 2000. *Simulation Modeling and Analysis*. third ed. McGraw-Hill, New York.
- Meng, G., Heragu, S., 2004. Batch size modeling in a multi-item, discrete manufacturing system via an open queueing network. *IIE Transactions* 36, 743–753.
- Mitrani, I., Puhalskii, A., 1993. Limiting results for multiprocessor systems with breakdowns and repairs. *Queueing System: Theory and Applications* 14, 293–311.
- Morrison, J.R., Martin, D.P., 2007. Practical extensions to cycle time approximations for the G/G/m-queue with applications. *IEEE Transactions on Automation Science and Engineering* 4 (4), 523–532.
- Myers, R.H., Montgomery, D.C., 2002. *Response Surface Methodology: Process and Product Optimization Using Designed Experiment*. second ed., Wiley-Interscience.
- Piepel, G.F., 1988. Programs for generating extreme vertices and centroids of linearly constrained experimental regions. *Journal of Quality Technology* 20, 125–139.
- Psichogios, D., Ungar, L., 1994. SVD-NET: An algorithm that automatically selects network structure. *IEEE Transactions on Neural Networks* 5 (3), 513–515.
- Sabuncuoglu, I., Touhami, S., 2002. Simulation metamodeling with neural networks: an experimental investigation. *International Journal of Production Research* 40 (11), 2483–2505.
- Schömig, A., Fowler, J.W., 2000. Modelling semiconductor manufacturing operations. In: *Proceedings of the Ninth ASIM Simulation in Production and Logistics Conference*, Berlin, Germany, pp. 55–64.
- Seber, G.A.F., Wild, C.J., 2003. *Nonlinear Regression*. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Shantikumar, J.G., Ding, S., Zhang, M.T., 2007. Queueing theory for semiconductor manufacturing systems: a survey and open problems. *IEEE Transactions on Automation Science and Engineering* 4 (4), 513–522.
- Spence, A.M., Welter, D.J., 1987. Capacity planning of a photolithography work cell in a wafer manufacturing line. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Raleigh, NC, pp. 702–708.
- Tamura, S., Tateishi, M., Matsumoto, M., Akita, S., 1993. Determination of the number of redundant hidden units in a three-layered feedforward neural network. *Proceedings of 1993 International Joint Conference on Neural Networks* 1, 335–338.
- Teoh, E.J., Tan, K.C., Xiang, C., 2006. Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. *IEEE Transactions on Neural Networks* 17 (6), 1623–1629.
- Vellido, A., Lisboa, P.J.G., Vaughan, J., 1999. *Neural networks in business: a survey of applications (1992–1998)*. *Expert Systems with Applications* 17, 51–70.
- White, H., 1989. Learning in artificial neural networks: a statistical perspective. *Neural Computation* 1 (4), 425–464.
- Whitt, W., 1983. The queueing network analyzer. *Bell System Technology Journal* 62, 2779–2815.
- Witczak, M., 2006. Toward the training of feed-forward neural networks with the D-optimum input sequence. *IEEE Transactions on Neural Networks* 17 (2), 357–373.
- Xiang, C., Ding, S., Lee, T., 2005. Geometrical interpretation and architecture selection of MLP. *IEEE Transactions on Neural Networks* 16 (1), 84–96.
- Yang, F., Ankenman, B.E., Nelson, B.L., 2007. Efficient generation of cycle time-throughput curves through simulation and metamodeling. *Naval Research Logistics* 54, 78–93.
- Yang, F., Liu, J., Nelson, B.L., Ankenman, B.E., Tongarlak, M., *Metamodeling for cycle time-throughput-product-mix surfaces using progressive model fitting*. *Production Planning and Control*, accepted for publication.
- Zhang, G.P., 2007. *Avoiding Pitfalls in Neural Network Research*. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*. 37(1), 3–16 (J. Mack Robinson Coll. of Bus., Georgia State Univ., Atlanta, GA).