Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

DEVELOPING EFFICIENT SIMULATION METHODOLOGY FOR COMPLEX QUEUEING NETWORKS

Ying-Chao Hung

Graduate Institute of Statistics
National Central University
Chung-Li, Taiwan 320, R.O.C.

George Michailidis
Derek R. Bingham

Department of Statistics
The University of Michigan
Ann Arbor, MI 48109-1092, U.S.A.

ABSTRACT

Simulation can provide insight to the behavior of a complex queueing system by identifying the response surface of several performance measures such as delays and backlogs. However, simulations of large systems are expensive both in terms of CPU time and use of available resources (e.g. processors). Thus, it is of paramount importance to carefully select the inputs of simulation in order to adequately capture the underlying response surface of interest and at the same time minimize the required number of simulation runs. In this study, we present a methodological framework for designing efficient simulations for complex networks. Our approach works in sequential and combines the methods of CART (Classification And Regression Trees) and the design of experiments. A generalized switch model is used to illustrate the proposed methodology and some useful applications are described.

1 INTRODUCTION

Traditional network models have played an important role over the last four decades in providing insight in the dynamics of telecommunication, computer, manufacturing, traffic and distribution systems (Baccelli and Bremaud 1994; Serfozo 1999; Walrand 1988). Such complex stochastic systems typically have a large number of control parameters that can have a significant impact on the performance of the system. Typical performance measures are the system's throughput, backlogs and delays of customers/jobs, as well as other general costs/revenues associated with their operation. Over the years a fairly rich literature has been developed for determining the throughput capacity of complex systems, as well as constructing control policies that achieve maximum throughput under different stochastic assumptions on the input and service processes; for example, drift analysis (Hajek 1982), fluid models (Dai and Meyn 1994; Dai 1995; Meyn 1996), sample path analysis (Baccelli and Bremaud 1994; Walrand 1988). On the other hand, it is a much harder task to obtain analytical results for backlogs and delays, where one relies on stochastic comparisons, coupling arguments (Bambos and Michailidis 2002; Xia, Michailidis, Bambos, and Glynn 2001) and large deviation principles (Schwartz and Weiss 1994). However, one usually needs to impose strong *structural* assumptions about the system under study (e.g. balanced input rates). Therefore, simulation becomes an important tool for understanding the performance of complex queueing models.

Simulation can provide insight to the behavior of a complex system by identifying *response surfaces* for several performance measures, such as backlogs and delays as functions of various control parameters; e.g. control policies, distributions of inputs and service times, buffer sizes, etc. However, simulation of large systems are expensive both in terms of CPU time and use of available resources (e.g. processors). In this study, we develop methodology for designing efficient simulations for complex networks. From a modeling perspective, the goal is to (i) adequately capture the underlying response surface, (ii) efficiently use the available experiment runs, (iii) provide an interpretable model and (iv) provide a means to evaluate the "goodness" of the model. Our approach works in sequential and combines classification and regression trees (CART) (Breiman, Friedman, Olshen and Stone 1984) and the design and analysis of experiments (Federov 1972; Wu and Hamada 2000), while the first one is used to partition the input space into homogeneous subregions and the second one selects the optimal location of the inputs to simulate the system.

The paper is organized as follows. In Section 2 a specific queueing model is introduced and used to illustrate the proposed methodology. In Section 3, the proposed approach is presented for a small system and some applications are described. In Section 4 some concluding remarks are drawn and an alternative approach currently under investigation is briefly discussed.

2 A GENERALIZED SWITCH MODEL

In this Section we introduce a particular queueing model exhibiting complex dynamics that is used to motivate and illustrate the proposed methodology. Consider a queueing system consisting of Q infinite capacity first-in-first-out (FIFO) queues in parallel, each queue corresponding to a different class of job traffic. The class q jobs arrive according to a renewal process $\mathcal{A}_q$ of mean rate λ_q, $q = 1, \dots, Q$. There are also a pool of Q processors, whose service times are mutually independent random variables, independent of the arrival processes, with distributions that depend on both the job class and the processor class. That is, the processing time of a class q job assigned to a class j processor has distribution P_{jq} and mean $0 < \mu_{jq}^{-1} < \infty$. At certain points in time, a processor must decide whether or not to process a job, and if so, from which of the Q classes. This basic queueing model captures the essence of a fundamental resource allocation problem in many modern communication, computer and manufacturing systems involving heterogeneous processors and multiple classes of job traffic flows (Hung 2002).

The decision mechanism employed by the processors at each of their decision points defines a *processor allocation policy*. It can be seen that due to the fact that all processors can process all classes of jobs, with *varying degrees of efficiency*, any processor allocation scheme induces complex dynamics.

The most fundamental problem for any queueing system is that of system stability. For a given set of service rates $\{\mu_{jq}\}$ and mean arrival vector $\vec{\lambda} = (\lambda_1, \dots, \lambda_Q)$, the system is said to be *stable* under some policy π, if the conditional expectation $E[\vec{N}^{\pi}(t)|\vec{N}^{\pi}(0)]$ is uniformly bounded above by a constant, where $\vec{N}^{\pi}(t)$ is the vector of queue lengths at time t (Hajek 1982). The set of arrival rate vectors for which the queueing system under study *may* be stabilizable by some policy π is given by the set (Hung 2002):

$$\mathcal{S} = \{\vec{\lambda} \in \mathbb{R}_+^Q : \lambda_q < \sum_{j=1}^{Q} w_{jq}\mu_{jq}\} \tag{1}$$

where $w_{jq} \in [0, 1]$ and satisfy $\sum_{q=1}^{Q} w_{jq} \leq 1$. The region $\mathcal{S}$ is known in the queueing literature as the *stability region* of the system (Walrand 1988). In Figure 1, the region $\mathcal{S}$ is shown for a system with 2 job classes and 2 processors (called henceforth a 2×2 system).

It can be shown that if the average input rate $\vec{\lambda} \notin \mathcal{S}$, then for any policy π, $\lim_{t\to\infty} \vec{N}^{\pi}(t) = \infty$. In words, this implies that there is *no* scheduling policy π that can keep the queue backlogs (and consequently job delays) finite in the long run (Hung 2002). On the other hand, if $\vec{\lambda} \in \mathcal{S}$ it does *not* necessarily imply that $E[\vec{N}^{\pi}(t)|\vec{N}^{\pi}(0)] < \infty$ for any server allocation policy π. However, a particular policy called Maximum Weighted Queue Length (MWQL) that at any decision point in time allocates processor j to the queue with maximum product $N_q^{\text{MWQL}}(t)\mu_{jq}$ maximizes the system's throughput; that is, $\sup_{t>0} E[\vec{N}^{\text{MWQL}}(t)] < \infty$ as long as $\vec{\lambda} \in \mathcal{S}$ (Hung 2002).

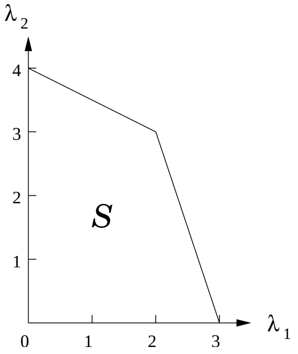

Figure 1: The Stability Region $\mathcal{S}$ of a 2 Queue, 2 Processor System with $\mu_{11} = 2$, $\mu_{12} = 1$, $\mu_{21} = 1$, $\mu_{22} = 3$

The question of interest then becomes how other performance measures, such as job delays, behave under the MWQL policy, for different choices of the arrival and service time distributions (e.g. exponential, gamma, etc). In order to answer this question one has to resort to a simulation study.

3 METHODOLOGY FOR SIMULATION

The complex relationship between the system response and the inputs requires a flexible model to adequately approximate the network dynamics. One could use a non-parametric regression approach (e.g. MARS (Friedman 1991)) to help model the response. However, non-parametric procedures do not provide an easy to interpret model relating the input factors to the response. Moreover, the theory of which input points to choose in the $\mathcal{S}$ region in order to fit a model with good properties (e.g. small standard errors of the coefficients) has not been developed. The aim of this endeavor is to find an easily interpretable response surface where we can gain insight into which factors dominate the response in different areas of the stability region $\mathcal{S}$. Consequently, a polynomial based function is attractive. The proposed approach is based on fitting treed models (Alexander and Grimshaw 1996). A treed model is an elaboration of conventional regression tree models (Breiman, Friedman, Olshen, and Stone 1984) that use binary trees to partition data into homogeneous subsets, where the response can be described by a simple mean. Although such models can provide a useful approach for handling interactions and nonlinearities, they do not fully exploit partitions with more substantial structure within the subsets. To overcome this limitation, treed models are constructed so that the *model structure*, as opposed to the data, is homogeneous within a terminal node, which in turn allows a richer and more parsimonious overall model. Moreover, the theory of choosing the optimal location of the inputs in $\mathcal{S}$ so as to optimize the desired characteristics of polynomial regression models has been well studied over the years (Federov 1972; Wu and Hamada 2000). In our approach we use the D-optimality criterion (Federov 1972) to help decide which simulation trials to perform. Experiment designs based on the D-optimality criterion select the input points to minimize the overall uncertainty of the estimated regression coefficients for a polynomial regression model.

The proposed methodology is sequential and allows for efficient use of available resources and model evaluation at each step. The final result is a set of local linear models approximating the response throughout the stability region. The steps of the new methodology are summarized below:

1. Run a small simulation experiment based on an initial experiment design.
2. Perform a CART analysis on the results in step 1. This partitions the input space $\mathcal{S}$ into subregions $\mathcal{S}_1$, ..., $\mathcal{S}_k$.
3. In each subregion, $\mathcal{S}_i$, augment the existing simulation trials with additional trials so that a polynomial regression model of the desired degree can be fit. The additional design points should be allocated using an optimality criterion such as D-optimality.
4. Fit local polynomial regression models and evaluate the fit. For each subregion where the model is not adequate, increase the degree of the polynomial regression model and return to step 3.

In the first step of this procedure, an initial allocation of design points (experimental trials) is used. The purpose of this step is to have an initial set of trials so that CART can be used to partition the input space into more homogeneous subregions. This design could be based on a D-optimal design for a global second order model, a space filling design (Box and Draper 1987; Wu and Hamada 2000), expert knowledge or some combination of these. In practice, we have found that an initial allocation of about 25% of the experimental budget (number of simulations available) is a good rule of thumb for the size of the design in this step.

In step 2, a CART model is used to summarize the responses. This amounts to a partition of the input region, $\mathcal{S}$, into more homogeneous subsets so local polynomial models can be fit. Because the CART procedure will likely result in some partitions where there are too few degrees of freedom to fit the polynomial models, the existing experiment trials in each subregion are augmented with additional design points following the D-optimality criterion. By augmenting the already existing trials with only enough trials necessary to fit the regression model, one is able to efficiently use the available resources.

Next, a polynomial regression model is fit within each subregion and the goodness of fit is evaluated. The local model evaluation is done using the usual residual analysis techniques. When the model fit is not adequate, the degree of the polynomial is increased and more trials are performed. This step allows the resources to be used in regions where the model is most complex.

3.1 Illustration of Methodology Applied to a 2×2 System

We first focus our attention to a system comprised of two job classes and two processors with service rates $\mu_{11} = 2, \mu_{12} = 1, \mu_{21} = 1, \mu_{22} = 3$. The stability region $\mathcal{S}$ of this system was shown in Figure 1. Our main objective is to approximate as well as possible the delay surface of interest (e.g. average delay, median delay, 95th percentile) of this system over the entire stability region $\mathcal{S}$ with the smallest possible number of simulation runs.

A naive, but popular in practice, approach is to superimpose on $\mathcal{S}$ a regular grid of sufficient density and then simulate the system at all the grid points. The resulting average delay surface under Poisson arrival processes and exponential service times based on 60 simulation runs is shown in Figure 2. The graph reveals that the average delay is small for lightly loaded systems, and becomes an order of magnitude larger at the 3 corners of $\mathcal{S}$. Moreover, the picture in the right panel of Figure 2 suggests that it would be beneficial to use the logarithm of the average delay (or any other metric of interest) as the response variable in the model building stage.

Remark: This naive approach becomes impractical for systems with higher dimensional input spaces. For example, consider generalized switches with more than 3 job classes. In those cases, the number of grid input configurations that need to be simulated increases exponentially, and because of constraints in computational resources one may need to use a very sparse grid that would fail to capture the underlying response surface.

The fitted surface based on 60 simulation runs would serve as a benchmark for testing the performance of the proposed methodology. Let $y(\vec{\lambda})$ denote the response variable (in this case the logarithm of the average delay) for the input configuration $\vec{\lambda}$. Our objective is to construct a statistical model of the form

$$y(\vec{\lambda}_i) = \sum_{k=0}^{K} \beta_k f_k(\vec{\lambda}_i) + \epsilon_i, \;\; i = 1, \ldots, n, \tag{2}$$

for appropriately chosen functions f_k, and where ϵ is a random error term, by using as few observations (that are going to be obtained by simulating the queueing system) as possible (small n). The number of simulation trials,

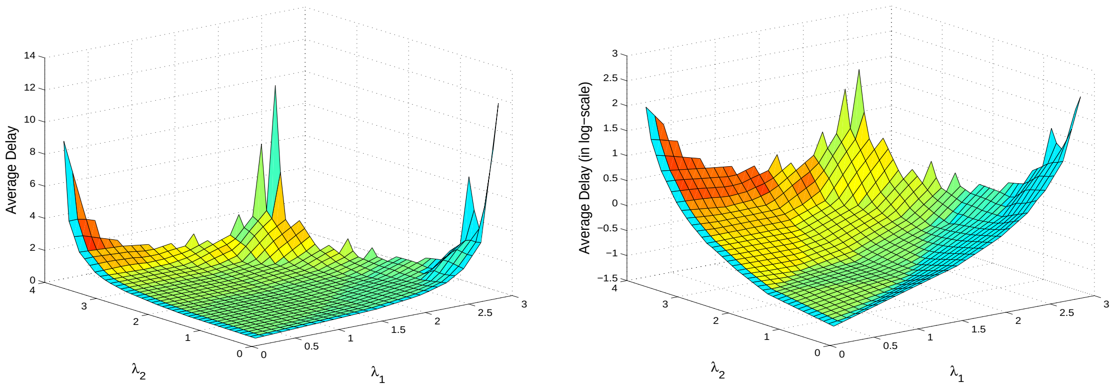

Figure 2: Linearly Interpolated Average-Delay Surface (Left Panel) and Average-Delay Surface in Log-Scale (Right Panel) for the 2×2 Model with Poisson Arrivals and Exponential Service Times under the MWQL Policy

n, is largely determined by the computational resources available, system complexity and the degree of accuracy we are interested in achieving. Then, the main issue becomes which input configurations $\vec{\lambda}_i$ to choose; i.e. which locations in the $\mathcal{S}$ region to select for simulating the system. In our methodology the functions $f_k(\cdot)$ correspond to polynomials, usually of low degree (e.g. 1 and 2).

We begin by considering the second order model that involves linear and quadratic main effect terms and linear interactions among the factors over the entire feasible region. So, for the 2×2 system, the model is:

$$y(\vec{\lambda}) = \beta_0 + \beta_1\lambda_1 + \beta_2\lambda_2 + \beta_3\lambda_1^2 + \beta_4\lambda_2^2 + \beta_5\lambda_1\lambda_2 + \epsilon. \quad (3)$$

The use of quadratic terms here is motivated by expert knowledge about the behavior of queueing systems and reinforced by the patterns observed in Figure 2.

In order to estimate the parameters of this model, a minimum of 7 simulation runs are needed (6 for the regression coefficients and 1 for the variance of the error term). The mathematical theory of D-optimality specifies that the optimal location of these 7 input configurations should be those given in Figure 3. The idea behind D-optimal design is to choose the locations in such a way, so as to minimize the volume of the confidence ellipsoid for the regression coefficient vector $\vec{\beta}$ (Federov 1972).

Based on these 7 design points, the resulting predicted average delay surface (after interpolation) and the plot of the difference of the predicted delays according to the model and the true responses (derived directly from simulation) for 60 input configurations are shown in Figure 4. It can be seen that this model approximates fairly well the “true” underlying response surface. However, as the residual plot reveals and as can be gleaned from the picture, it significantly overestimates the average delay for lightly loaded system (near the origin), while underestimating it close to the boundaries of $\mathcal{S}$. The shortcoming of this modelling approach is that the model is not flexible enough for this type of response surfaces we are interested in.

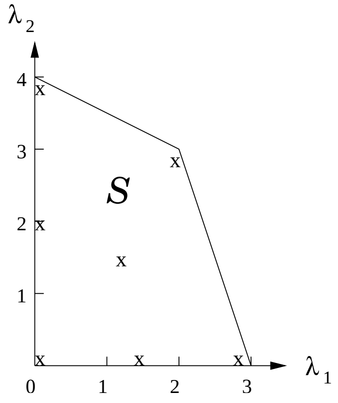

Figure 3: The 7 Input Locations Selected by D-Optimal Design Based on Model (3)

Therefore, we use a more flexible approach that utilizes the insight gained from model (3) in our methodological framework. In general, one can anticipate the general patterns observed in Figure 2 and Figure 4. That is, when the input rates are low, the average delay will be fairly small. It is only when the queues get backed up will the average delays get large, and the pattern near the boundary will be observed. Therefore the second order model in (3) is a good starting point for investigation. As a consequence, Step 1 of our approach uses the same design points for a second order model as above. Next in Step 2, a conventional tree regression model is then fit to the responses of the 7 design points and according to its results, the $\mathcal{S}$ region is subdivided into 3 more homogeneous subsets $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ (see Figure 5).

In Step 3 and Step 4, we need to decide what type of model to fit in the resulting 3 subregions and then allocate additional design points (always according to the D-optimality

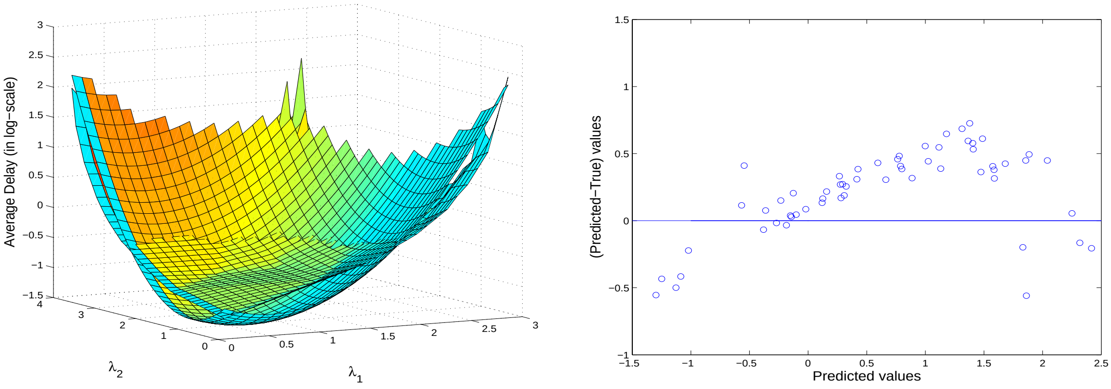

Figure 4: The Predicted Average-Delay Surface (in Log-Scale) Based on 7 Design Points Using Model (3) versus the "True" Surface Derived from the Naive Approach (Left Panel) and the Plot of (Predicted-True) Delays in Log-Scale (Right Panel)

criterion used) by *augmenting* the existing design (i.e., add points to the already available ones within every subregion $\mathcal{S}_i$, $i = 1, 2, 3$). We start by fitting a simple polynomial model and the fit is evaluated within each subregion using standard regression techniques. Next, for subregions where the model fit is not adequate only, sequentially increase the degree of each polynomial, adding the appropriate number of design points to fit the model and keeping 3 degrees of freedom for error. After a recursive running of Step 3 and Step 4, model (3) was fit to each of the subregions and the resulting surface and its residual plot are given in Figure 6.

It can be seen from Figure 6 that this model approximates very well the 'true' response surface, especially at the corners of the $\mathcal{S}$ region where most of the action in terms of average delay is. It is worth noting that this model uses only 27 (3 subregions × 9 points/subregion) simulation runs. The resulting local polynomials for the three subregions $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ are listed below:

$$\left\{ \begin{array}{l} \log(y(\vec{\lambda})) = -1.15^* + 0.08^*\lambda_1 - 0.15^*\lambda_2 - 0.13^*\lambda_1\lambda_2 \\ \qquad\qquad + 0.26^*\lambda_1^2 + 0.19^*\lambda_2^2; \;\; R^2 = .992 \\ \log(y(\vec{\lambda})) = 19.19 - 5.05\lambda_1 - 13.70^*\lambda_2 + 1.22\lambda_1\lambda_2 \\ \qquad\qquad + 1.24^*\lambda_1^2 + 2.42^*\lambda_2^2; \;\; R^2 = .940 \\ \log(y(\vec{\lambda})) = 56.38^* - 44.98^*\lambda_1 - 0.41\lambda_2 - 0.22\lambda_1\lambda_2 \\ \qquad\qquad + 9.11^*\lambda_1^2 + 0.51^*\lambda_2^2; \;\; R^2 = .991 \end{array} \right.$$

where $*$ denote the significant coefficients at the 10% level and with all three models being statistically significant at the 1% level. Based on the above local polynomial models, we can interpret how the input factors (input arrival rates) affect the average system delay in each subregion. For example, the relatively large (in magnitude) coefficients of the last two polynomials indicate that a slight change of input rates in $\mathcal{S}_2$ and $\mathcal{S}_3$ can significantly affect the average system delay. On the other hand, smaller (in magnitude) coefficients of the first polynomial constitute a relatively flat average delay surface on $\mathcal{S}_1$. Note that only the coefficient of the term $\lambda_1\lambda_2$ in the first polynomial is statistically significant. This implies that the interaction between two input rates might play a much more important role that affects the average system delay in $\mathcal{S}_1$.

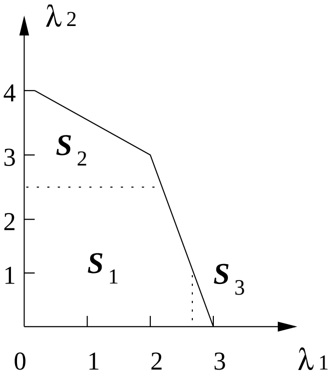

Figure 5: Three Subregions Derived from CART Based on the Responses of the 7 Design Points

3.2 Application 1: Predictions

The main benefit of having a good model for the delay response surface is that we can predict the average system delay at *untried* input locations. To investigate the quality of our predictions based on the constructed treed model, we compare the true responses obtained from simulating the system at m randomly chosen locations with the predictions obtained from the model. In order to carry out this comparison the criteria of empirical integrated squared error (EISE) and maximum absolute error (MAE) are used (Wu and Hamada 2000). The results for the predictions at 30 randomly chosen locations in the $\mathcal{S}$ region, for the 2×2 model under exponential interarrival and service times for the treed model, the original quadratic model (3) and a

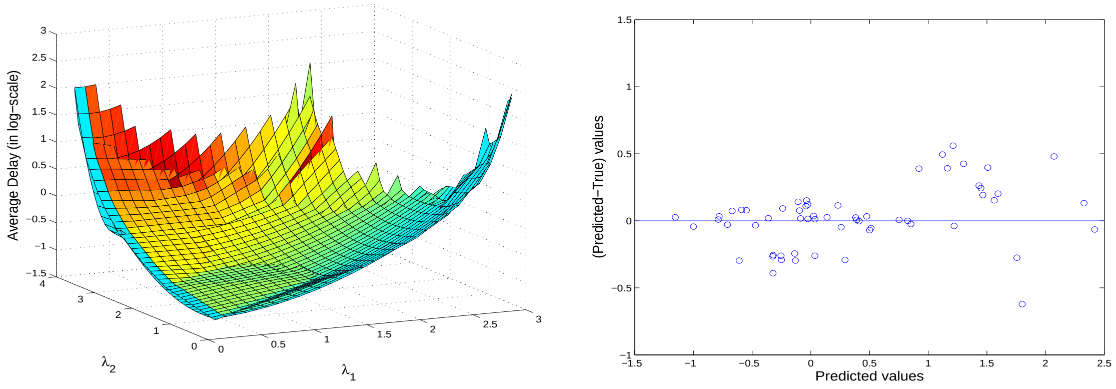

Figure 6: The Predicted Average-Delay Surface (in Log-Scale) Based on the Flexible Approach versus the "True" Surface Derived from the Naive Approach (Left Panel) and the Plot of (Predicted-True) Delays in Log-Scale (Right Panel)

third competing approach that fits multivariate adaptive regression splines (MARS) (Friedman 1991) to the responses at the 27 locations used in the treed models under these two criteria are given in Table 1. It can be seen that our flexible approach has the smallest EISE (0.218^2) and MAE (0.512) relative to a data range 2.406 (maximum true value - minimum true value). It clearly outperforms the other two approaches.

Table 1: The Comparison of EISE and MAE for Three Different Approaches Based on 30 Predictions under the MWQL Policy

	Response: log(average delay)	
	EISE	MAE
Model (3)	$(0.445)^2$	0.699
MARS	$(0.724)^2$	1.520
Treed Model	$(0.218)^2$	0.512
Data Range	2.406	

3.3 Application 2: Comparisons of Response Surfaces

Another benefit of the proposed methodology is that it allows one to make comparisons between different response surfaces that correspond to the system under study for different control policies or for the same control policy but under different stochastic assumptions. We illustrate this aspect of our methodology by looking at our generelized switch operating under another policy called the Maximum Service Rate (MSR) policy, that at decision times allocates server j to the job class q with maximum service rate μ_{jq}, with the assumption of exponential interarrival and service times.

First, applying our methodology a good model for the average delay is built. At the second stage, using the derived treed models the predicted average delay (in log-scale) is calculated at 806 locations in $\mathcal{S}$ located on a 2-dimensional grid. Then, the differences of the delays for the system operating under two different policies are calculated and the result (in the form of contour plot) is given in Figure 7. It can be seen that the MSR policy outperforms the MWQL policy for lightly loaded systems and in the corners of the stability region, where almost all the incoming jobs belong to one of the two classes. It is easy to see that having one of the servers dedicated to the heavily loaded class and only the second server switching between classes (what is basically hapenning in those corners under the MSR policy) definitely helps with respect to average delay.

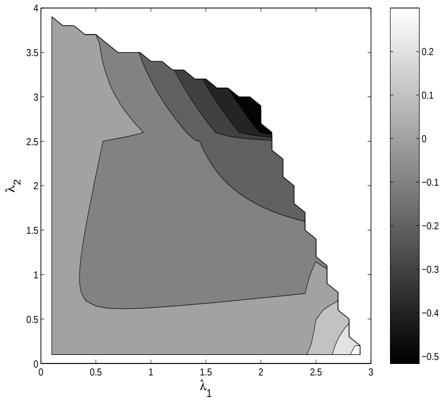

Figure 7: The Contour Plot of the Differences of the Average Delays (in Log-scale) for the System Operating under Policy MWQL and MSR (MWQL-MSR)

3.4 Application 3: Identify the Iuput Region for a Given Threshold of Response

Building a good model for the response surface over the entire $\mathcal{S}$ region also allows us to be able to identify a subset of all possible input rates $\mathcal{S}^U \subset \mathcal{S}$ when a threshold of the response is set, that is, $\mathcal{S}^U = \{\vec{\lambda} : y(\vec{\lambda}) < U, \vec{\lambda} \in \mathcal{S}\}$ for a given value U. For example, in the 2×2 system if the threshold of the average system delay (in log-scale) is set to be 1, all possible input rates $\vec{\lambda}$ satisfying that $y(\vec{\lambda}) < 1$ can then be solved by using the three local polynomials derived from our approach. The resulting input rate subset $\mathcal{S}^U$ is shown in Figure 8. It should be noted that the MWQL policy then provides a quarantee of Quality of Service (QoS) for all input rates $\vec{\lambda}$ that belong to this subset $\mathcal{S}^U$.

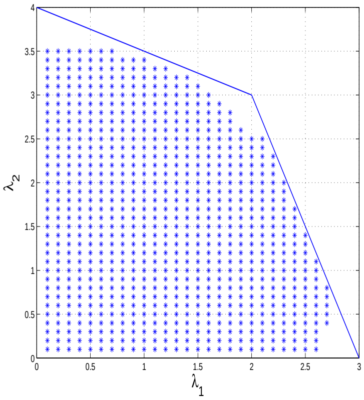

Figure 8: The Identified Input Rate Subset $\mathcal{S}^U$ (the Area with Notation "*") for a Given Threshohold $U = 1$ under the MWQL Policy

4 CONCLUDING REMARKS AND FUTURE WORK

We have presented a methodological framework for designing efficient simulations of complex stochastic networks. The proposed framework follows a sequential approach that uses an initial design in order to partition the stability region ($\mathcal{S}$) of the system into homogeneous subregions and then fits different regression models to each subregion. This methodology can easily handle systems with high dimensional input spaces, while the underlying theory for optimal design for MARS type of models is not available (we also have the results for the 3×3 and 8×8 systems). We are currently investigating computational issues that arise when one is dealing with very high dimensional spaces and also how to best compare the performance of different control policies.

Furthermore, we are also exploring an alternative methodology which is motivated by ideas in the field of computer experiments (Currin, Mitchell, Morris, and Ylvisaker 1991; Sacks, Welch, Mitchell, and Wynn 1989), and have obtained some initial promising results. This approach is also sequential in nature and is expected to be able to handle a more general type of response surfaces that might exhibit non-monotone properties.

REFERENCES

Alexander, W. P., and S. D. Grimshaw. 1996. Treed regression. *Journal of Computational and Graphical Statistics* 5: 156-175.

Baccelli, F., and P. Bremaud. 1994. *Elements of queueing theory*. Berlin: Springer-Verlag.

Bambos, N., and G. Michailidis. 2002. On parallel queueing with random server connectivity and routing constraints. *Probability in the Engineering and Information Sciences* 16: 185-203.

Box, G. E. P., and D. R. Draper. 1987. *Empirical model building and response surfaces*. New York: John Wiley & Sons.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and regression trees.* Monterey: Wadsworth.

Currin, C., T. Mitchell, M. Morris, and D. Ylvisaker. 1991. Bayesian prediction of deterministic functions with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association* 86: 953-963.

Dai, J. G., and S. P. Meyn. 1994. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Transactions on Automatic Control* 40: 1889-1904.

Dai, J. G. 1995. On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability* 5: 49-77.

Federov, V. V. 1972. *Theory of optimal experiments.* New York: Academic Press.

Friedman, J. 1991. Multivariate adaptive regression splines. *Annals of Statistics* 19: 1-141.

Hajek, B. 1982. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied Probability* 14: 502-525.

Hung, Y. C. 2002. *Modeling and analysis of stochastic networks with shared resources*. Ph.D. thesis, Department of Statistics, The University of Michigan, Ann Arbor, Michigan.

Meyn, S. P. 1996. Stability and optimization of queueing networks and their fluid models. *Proceedings of the Summer Seminar on "The Mathematics of Stochastic Manufacturing Systems"*, 17-21.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. Design and analysis of computer experiments. *Statistical Science* 4: 409-423.

Schwartz, A., and G. Weiss. 1994. *Large deviations for performance analysis, queues, communication and computing*. New York: Chapman and Hall.

Serfozo, R. 1999. *Introduction to stochastic networks*. New York: Springer.

Walrand, J. 1988. *Introduction to queueing networks*. Englewood Cliffs: Prentice Hall.

Wu, C. F. J., and M. Hamada. 2000. *Experiments: Planning, analysis, and parameter design*. New York: Wiley.

Xia, C., G. Michailidis, N. Bambos, and P. Glynn. 2001. Optimal control of parallel queues with batch service. *Probability in the Engineering and Information Sciences* 16: 289-307.

AUTHOR BIOGRAPHIES

YING-CHAO HUNG is an Assistant Professor of the Graduate Institute of Statistics at National Central University in Taiwan. He received the Ph.D. degree in statistics from the University of Michigan, Ann Arbor, in 2002. He is a member of ASA and INFORMS. His e-mail address is `<hungy@stat.ncu.edu.tw>`.

GEORGE MICHAILIDIS is an Assistant Professor of the Department of Statistics at the University of Michigan, Ann Arbor. He is also an associate editor of *Journal of Computational and Graphical Statistics.* His research interests include multivariate analysis, computational statistics, bioinformatics, stochastic processing networks and network tomography. You can reach him by e-mail at `<gmichail@umich.edu>`.

DEREK R. BINGHAM is an Assistant Professor of the Department of Statistics at the University of Michigan, Ann Arbor. His research interests include industrial statistics, design of experiments, optimal design and statistical computing. You can reach him by e-mail at `<dbingham@umich.edu>`.