*Proceedings of the 1999 Winter Simulation Conference*
*P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.*

# THE DEVELOPMENT OF A METHODOLOGY FOR THE USE OF NEURAL NETWORKS AND SIMULATION MODELING IN SYSTEM DESIGN

Mahdi Nasereddin
Mansooreh Mollaghasemi

Department of Industrial Engineering and Management Systems
University of Central Florida
Orlando, FL 32826, U.S.A.

## ABSTRACT

In this paper the use of metamodels to approximate the reverse of simulation models is explored. This purpose of the approach is to achieve the opposite of what a simulation model can do. That is, given a set of desired performance measures, the metamodels output a design to meet management goals. The performance of several neural network simulation metamodels was compared to the performance of a stepwise regression metamodel in terms of accuracy. It was found that in most cases, neural network metamodels outperform the regression metamodel. It was also found that a modular neural network performed the best in terms of minimizing the error of prediction.

## 1 INTRODUCTION

System design/redesign is a complex process in which models are used to make decisions on changes to existing or proposed systems. The goal of the design process is to design a system that meets or exceeds certain performance measures without violating any constraints. Simulation modeling is one of the most popular tools for the design and analysis of complex systems. This popularity is due to its flexibility, its ability to model systems more accurately, and its ability to model the time dynamic behavior of the system. With simulation modeling, however, the relationships between the design parameters and performance measures are not explicitly known. Therefore, system design using simulation becomes a trial and error process in which a set of design parameters are used in the simulation model to predict a set of performance measures. If the performance measures are acceptable, a good design has been identified, otherwise the process is repeated until a satisfactory set of performance measures is achieved. Because of the iterative nature of this procedure, this process can be time consuming and expensive.

In order to overcome this problem, researchers have proposed the use of metamodels. The main objective of a simulation metamodel is to accurately represent the relationship between inputs and outputs over wide ranges of interest, and to be more computationally efficient than simulation (Kilmer, Smith, and Shuman 1997). If the simulation runs are time-consuming and expensive, the advantages of using a metamodel are evident. After the metamodel has been built, there may not be a need to run the expensive and time consuming simulations, thus providing a quick way of answering "what if " type of questions.

Two approaches have been used for developing simulation metamodels: the direct simulation metamodeling approach, and the reverse simulation metamodeling approach (Figure 1). When building the metamodel using the direct approach, the inputs of the simulation (design parameters) are used as inputs for the metamodel, and the outputs of the simulation (performance measures) are used as desired outputs for the metamodel. When building a reverse simulation metamodel, the outputs of the simulation (performance measures) are used as inputs to the metamodel, and the inputs of the simulation (design parameters) are used as desired outputs of the metamodel. The advantage of using a reverse simulation metamodel as a design tool is that the process is no longer iterative. The decision-maker inputs the required performance measures and the reverse metamodel outputs the necessary parameters to achieve those measures. The graphical representation of both direct and reverse metamodeling is shown in Figure 1.

The objective of this paper is to develop a methodology for using simulation and neural networks to build a reverse-simulation metamodel that will be used as a decision support tool when designing a new system or redesigning an existing one. This decision tool will suggest the system's design parameters when the required performance measures are specified. In this paper several

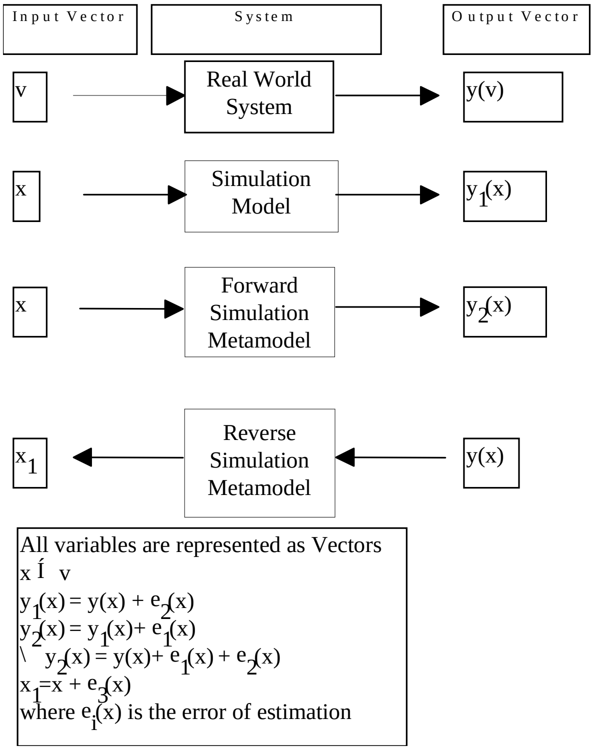
Input Vector
System
Output Vector
v
Real World System
y(v)
x
Simulation Model
$y_1(x)$
x
Forward Simulation Metamodel
$y_2(x)$
$x_1$
Reverse Simulation Metamodel
y(x)
All variables are represented as Vectors
x Í v
$y_1(x) = y(x) + e_2(x)$
$y_2(x) = y_1(x)+ e_1(x)$
\ $y_2(x) = y(x)+ e_1(x) + e_2(x)$
$x_1=x + e_3(x)$
where $e_i(x)$ is the error of estimation

Figure 1: Graphical and Mathematical Representation of Direct and Reverse Simulation Metamodels

neural network topologies are investigated and compared to a stepwise regression metamodel. The performance measure used is the normalized error of prediction.

## 2 RELATED WORK

There has been many neural networks Applications in the Industrial Engineering area. Ramesh Sharda (1998) summarized what has been done in the Operations Research field until 1996. He referenced more than 140 papers using neural networks in industrial engineering applications. Much work has also been done in the area of simulation and artificial intelligence. Oren (1994) referenced 198 papers in the application of artificial intelligence and simulation. But most of the work done was in the knowledge-based systems and simulation. Within the area of simulation and neural networks two different areas of simulation metamodels have surfaced in the last decade: The direct simulation metamodeling, and the reverse simulation metamodeling. In this paper, we will only discuss the reverse simulation metamodels. For direct simulation metamodels, the reader is referred to Kilmer, Smith, and Shuman (1997) and Badiru, and Sieger (1996).

The use of neural networks as a metamodeling technique to do the reverse of simulation modeling has been reported in three papers (Mollaghasemi, LeCroy, and Georgiopoulos 1998; Chyssolouris, G. Lee, M., and Domroese, M. 1990; Chryssolouris, G. and Domroese, M. 1991).

Chyssolouris, Lee, and Domroese (1990) and Chyssolouris and Domroese (1991) explored the use of neural networks for identifying the relative importance of pertinent manufacturing criteria for given performance measures. The simulation model used in both papers is a job shop. The simulation was performed five times, each time with a different job shop configuration. A neural network that used the generalized delta rule was trained using the five simulation runs. The performance measures were inputted into the neural network and the network was trained to achieve the job shop configuration associated with those performance measures. Chyssolouris and Domroese compared the neural network performance to that of a first order linear regression. It was found that the neural network outperformed the first order linear regression.

Mollaghasemi, LeCroy, and Georgiopoulos (1998) applied a neural network metamodel to a real world application involving the test operations of a major semiconductor manufacturing plant. Given a set of desired performance measures in terms of cycle time, WIP, and utilization of three different testers, the metamodel suggested a suitable design in terms of scheduling rules, and the number of each type of tester to achieve these objectives. The results of the metamodel were validated by comparing them with the results obtained from the simulation model. The authors reported encouraging results.

## 3 EXPERIMENTAL DESIGN

In order to demonstrate the effectiveness of using a reverse metamodel as a decision support tool, a simulation of a simple re-entrant manufacturing model with five machine cells running three different parts was created. Each part goes through the following machine cell sequence: 1-2-3-4-1-2-3-4-1-2-3-4-5 before exiting the system. Exponential processing times were used to model the processing times. The mean processing times for each part can be found in Table 1.

Table 1: The Mean Processing Times (in Minutes) for the Re-Entrant Model

| | **Cell 1** | **Cell 2** | **Cell 3** | **Cell 4** | **Cell 5** |
|---|---|---|---|---|---|
| **Part A** Level 1 | 10 | 8 | 14 | 12 | |
| Part A Level 2 | 12 | 9 | 14 | 11 | |
| Part A Level 3 | 14 | 10 | 14 | 10 | 20 |
| **Part B** Level 1 | 9 | 9 | 14 | 13 | |
| Part B Level 2 | 11 | 10 | 14 | 12 | |
| Part B Level 3 | 13 | 11 | 14 | 11 | 20 |
| **Part C** Level 1 | 11 | 7 | 14 | 13 | |
| Part C Level 2 | 13 | 8 | 14 | 12 | |
| Part C Level 3 | 15 | 9 | 14 | 11 | 20 |

The following is the list of inputs and outputs of the simulation:

INPUTS

1 Interarrival time: two levels were used (20 or 30 minutes)
2 Number of machines in cells 1, 2, 3, and 4: three levels were used in each cell (2, 3, or 4 machines)
3 Number of machines in cell 5: two levels (1 or 2 machines)
4 Scheduling Policy: three levels were used (First In First Out, lowest processing time, or the part that is closest to completion)

OUTPUTS

1 Machine utilization of each of the five machine cells (5 output variables)
2 Cycle time of each of the three parts (3 output variables)
3 Average work in process inventory (WIP) ( 1 output variable)

Although this model is comparatively simple when compared with real world models, the experimental design space of the above model consists of 972 data points. Therefore, in a real life situation the number of possible input combinations could be much larger. The computational time is still magnified by the fact that multiple replications of each simulation run are required.

### 3.1 Neural Network Metamodels

Several neural network topologies were investigated:

1 Backpropagation neural network with a sigmoidal activation function using the delta rule learning algorithm.
2 General regression neural network which is a general purpose network paradigm used mainly for system modeling and prediction. Three different summation functions were used: Eucledian, city block, and projection summation function.
3 Modular neural network which consists of several backpropagation networks competing to learn different aspects of the problem. Four different learning rules were used: Quickprob, delta rule, delta bar delta, and maxprob.
4 Learning vector quantization which is a classification network which assigns vectors to one of several classes. It consists of a Kohonen layer which learns and performs the classification.
5 Radial basis function network which is a general purpose network paradigm used mainly for system modeling, prediction, and classification. Three different summation functions were used: Eucledian, city block, and projection summation function

A total of 10 different topologies were examined (1 Backprobagation, 3 general regression networks, 4 modular neural networks, 1 learning vector quantization, and 3 Radial basis function).

To build the metamodel, an orthogonal array experimental design consisting of 18 data points was used. For each setup 10 replication were made and the average of the 10 replications were computed. The input-output data set generated by the simulation was then used to train the metamodel. The outputs of the simulation (machine utilization of each of the five machine cells, cycle time of each of the three parts, and average work in process inventory (WIP)) were used as inputs when training the neural network metamodels. The inputs to the simulation (interarrival time, number of machines in cell 1, cell 2, cell 3, cell 4, and cell 5, and the scheduling Policy) were used as outputs when training the neural network Metamodels (Figure 2).

After training, the performances of the metamodels were evaluated using all the 972 data points. All the data points were used for evaluation to give a good understanding of the generalization capabilities of the metamodel. Generalization is the ability of the metamodel to predict the output of a set of inputs that it was not trained with. The mean square error of prediction was calculated for all the responses.

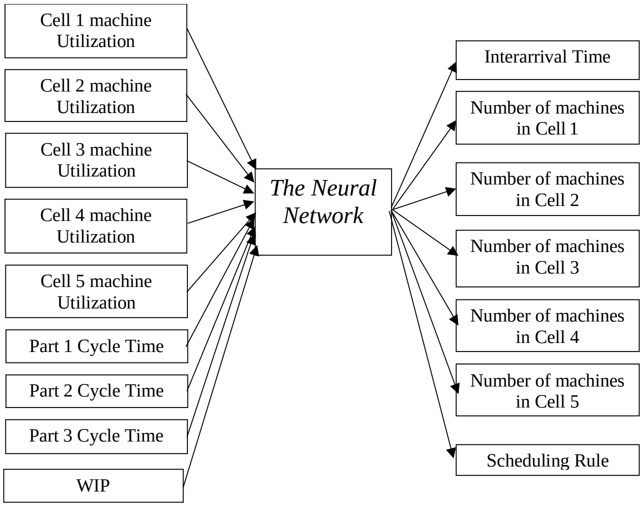
Cell 1 machine Utilization
Cell 2 machine Utilization
Cell 3 machine Utilization
Cell 4 machine Utilization
Cell 5 machine Utilization
Part 1 Cycle Time
Part 2 Cycle Time
Part 3 Cycle Time
WIP
The Neural Network
Interarrival Time
Number of machines in Cell 1
Number of machines in Cell 2
Number of machines in Cell 3
Number of machines in Cell 4
Number of machines in Cell 5
Scheduling Rule

Figure 2: The Reverse Simulation Metamodel

### 3.2 Regression Metamodel

The same data that was used for training the neural network metamodels was used to generate the regression metamodels. A stepwise regression was used to generate a linear approximation of the each of the controllable factors using the orthogonal array design of 18 data points. The outputs of the simulation were used as independent variables ($x_n$) when fitting the regression metamodels. The inputs to the simulation were used as dependent variables ($f(x_n)$) when fitting the regression Metamodels. To fit the regression metamodel a linear stepwise regression model was used. After fitting the regressions to the data points, the performance of the metamodels was evaluated using all the available 972 data points. The mean square error of prediction was calculated for all the responses.

## 4 RESULTS

It was found that the neural network metamodels (except for Linear Vector Quntization and Radial Basis Function using a Eucledian summation function) outperformed the stepwise regression metamodel. The modular neural network using the delta learning rule performed the best in terms of prediction accuracy (Figure 3). It was also found that for the modular neural network and the backpropagation neural network, the choice of the learning rule greatly affects the performance of the neural network (Figure 4). For the radial basis function, it was found that the network is not very sensitive for the choice to the summation function. The general regression neural network was more sensitive to the choice of the summation function (Figure 5).

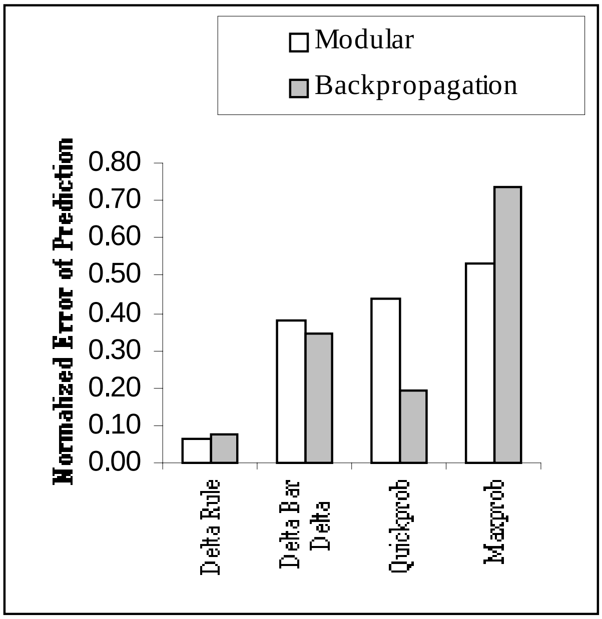

Figure 3: The Normalized Prediction Error for the Neural Network Metamodels and the Stepwise Regression Metamodel

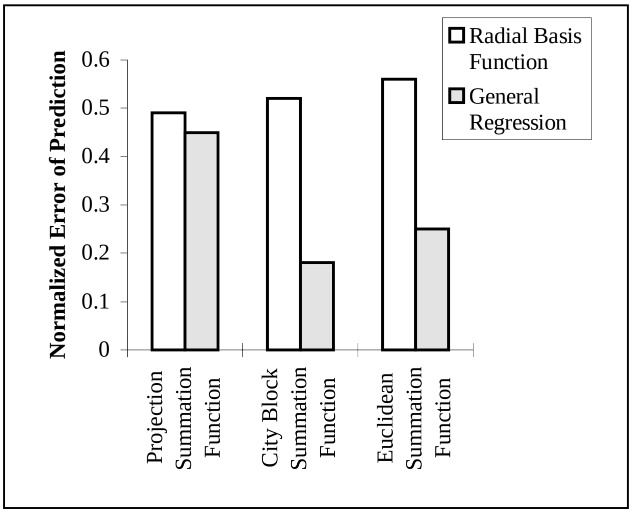

Figure 4: The Learning Rule Contribution to the Normalized Error of Prediction

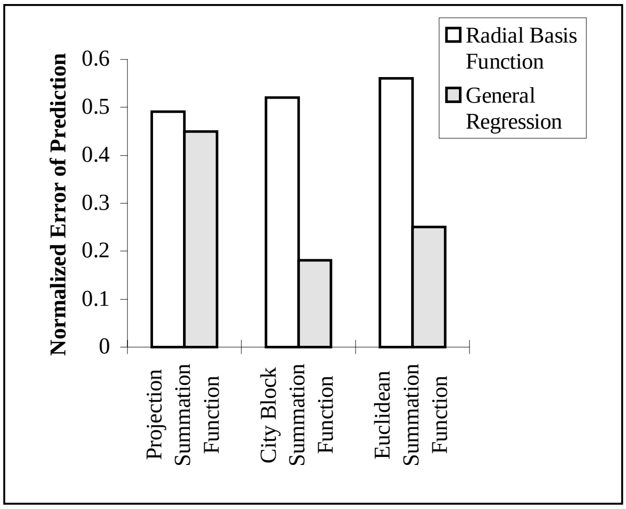

Figure 5: The Summation Function Contribution to the Normalized Error of Prediction for the General Regression and Radial Basis Function Neural Networks

None of the metamodels were able to predict the type of Scheduling Policy Used. The best correct classification was 51%. Increasing the number of training data points to 36 points improved the correct classification rate to 66%. Increasing the number of training points to 72 improved the correct classification rate to 71% (Figure 6). This indicates that the original number of data points used (18) was not enough to accurately predict the scheduling policy. Although, 18 points were enough to train the neural network to recognize the quantitative data (number of machines in each cell and interarrival time), they were not enough to recognize the qualitative data (type of scheduling policy). This suggests that two neural networks may be needed for developing the metamodels for this simulation model: one to predict the quantitative data, and the second to predict the qualitative data.

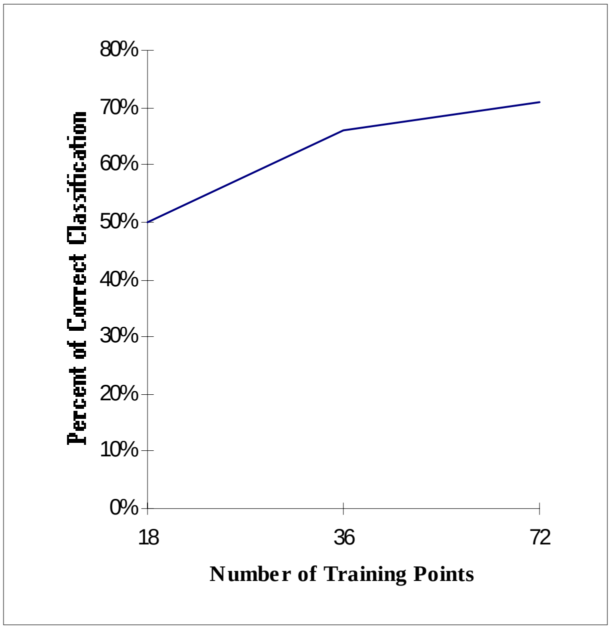

Figure 6: The Effect of Increasing the Number of Training Data on the Correct Scheduling Policy Classification

## CONCLUSIONS

The purpose of this research was to provide a methodology of how to build and examine a reverse simulation metamodel. Thus future research is needed to determine the best metamodel for other types of problems. Preliminary results show that neural network metamodels can outperform their regression counterparts.

Currently, our research is directed toward developing a methodology for building a neural network metamodel based on the type of design parameter, namely qualitative or quantitative, and the level of complexity of the solution surface. Our current research will provide a methodology for choosing the neural network training and testing data sets. The research will also provide a methodology for training the neural network.

## ACKNOWLEDGEMENT

This research was partially funded by Neural Ware Inc.

## REFERENCES

Badiru, Adedeji B. and Sieger David B., Neural Networks as Simulation metamodels in economic analysis of risky projects, *European Journal of Operations Research*, 1998, Vol. 105, n 1, 130-142

Barton, Russell R., Metamodels for Simulation Input-Output Relations, *Proceedings of the 1992 Winter Simulation Conference*, 289 - 299

Barton, Russell R., Simulation Metamodels, *Proceedings of the 1998 Winter Simulation Conference,* Vol. 1, 167 - 176

Choueiki, M. Hisham, Mount-Cambell, Clark A., and Ahalt, Stanley C., Building a Quasi Optimal Neural Network to Solve the short-term Load Forecasting Problem, *IEEE Transactions on Power Systems*, November, 1997, Vol. 12 no.4, 1432-1439

Choueiki, M. Hisham, and Mount-Cambell, Clark A., Training Data Development with the D-Optimality Criterion, *under review for IEEE Transactions of Neural Network*

Chryssolouris, G. and Domroese M., The use of neural networks in determining operational policies for manufacturing systems, *Journal of Manufacturing Systems*, 1990, Vol. 10 no. 2, 166-175

Chryssolouris, G. Lee, M., and Domroese M., Use of Neural Networks for the design of manufacturing systems, *American Society of Manufacturing Engineers*, 1991, Vol. 3 no 3, 187-194

Fishwick, Paul A., Neural network models in simulation: A comparison with traditional modeling approaches, *Proceedings of the 1989 Winter Simulation Conference*, 1989, 702-710

Hornik, Kurt, Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 1989, Vol. 2, 359-366

Hurrion, R., Using a Neural Network to Enhance the Decision Making Quality of a Visual Interactive Simulation Model, *Journal of Operations Research Society*, 1992. Vol. 43 no. 4, 333-341

Kelton, W. D., Sadowski, R. P., and Sadowski, D. A.,*Simulation with Arena*, McGraw-Hill 1998

Kilmer, Robert A., Application of neural networks to combat simulation, *Math Comp Modeling*, 1996, Vol. 23 no. 1, 91-99

Kilmer, Robert A., Smith, Alice E., and Shuman, Larry J. Neural Networks as a Metalmodeling technique for discrete event stochastic simulation, *under review for Computers & Industrial Engineerin.1998*

Kilmer, Robert A. Smith, Alice, and Shuman, Larry J., An emergency department simulation and neural network metamodel, *Journal of the society for health systems*, 1997, Vol. 5 no. 3, 63-79

Kilmer, Robert A. and Smith, Alice, Using Artificial Neural Networks to approximate a discrete event stochastic simulation model, *Intelligent Engineering Systems Through Artificial Neural Networks*, 1993, Vol. 3, 631-636

Kleijnen, J., Regression metamodels for generalizing simulation results, *IEEE transactions on systems, man, and cybernetics*, 1979, Vol. 9 no. 2, 93-96

Mollaghasemi M., Lecroy K., and Georgiopoulos M.. Application of neural networks and simulation modeling in manufacturing system design, *Interfaces*, Sep-Oct 1998

Oren, Tuncer, Artificial intelligence in simulation, *Annals of operations research*, 1994, Vol. 53, 287-319

Pierreval, H. and Huntsinger, R. An Investigation on neural network capabilities as simulation metamodels. *Proceedings of the 1992 Summer Computer Simulation*, 413-417

Sharda, Ramesh and Rampal, Rohit, Neural Networks and Management Science/Operations Research, *Encyclopedia of library and information science*, 1998, Vol. 61, 247-259

## AUTHOR BIOGRAPHIES

**MAHDI NASEREDDIN** is a Ph.D. candidate in the Department of Industrial Engineering at the University of Central Florida. He has a master's degree in Industrial Engineering from the same University. His research interests include simulation optimization, simulation metamodeling, project scheduling, and artificial intelligence especially neural networks and genetic algorithms. He is a member of Alpha Pi Mu, IIE and INFORMS.

**MANSOOREH MOLLAGHASEMI** is an associate professor of Industrial Engineering and Management Systems at the University of Central Florida. She received her Ph.D. in Industrial Engineering from the University of Louisville in 1991. She also holds a B.S. and an M.S. in Chemical Engineering from the same University. Her research interests involve simulation modeling and analysis, multiple response simulation optimization, and neural networks. She is a member of IIE and INFORMS.