Prediction and Control in a Just-In-Time Environment Using Neural Networks

by

Barry A. Wray

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Management Science

APPROVED:

Terry R. Rakes

Terry R. Rakes, Chairman

Edward R. Clayton

Edward R. Clayton

Loren Paul Rees

Loren P. Rees

Roberta S. Russell

Roberta S. Russell

Robert T. Sumichrast

April 13, 1992

Blacksburg, Virginia

Prediction and Control in a Just-In-Time

Environment Using Neural Networks

by

Barry A. Wray

(ABSTRACT)

The success of the Japanese just-in-time (JIT) with kanban inventory control technique has caused many manufacturing firms world-wide to implement similar systems in an attempt to remain competitive. Predicting and controlling the number of kanbans in an unstable environment is a complex decision involving many stochastic factors. This research investigates using neural computing (neural networks) to identify endogenous factors (shop conditions) and exogenous factors (product demand and supplier schedules) that are correlated with kanban system performance and to predict the optimal number of kanbans based on the "dynamic" interaction (changing over time) of these factors inherent in many production environments. The purpose of the research is to test the interpolative ability of a neural network to synthesize a multidimensional response surface from sample values and to perform factor screening on the inputs. First, a JIT shop simulator capable of utilizing different factor levels is used to generate data on shop performance for different kanban levels for 560 dynamic shop scenarios. Each combination of shop factor levels, along with the corresponding optimal number of kanbans, is saved in a data file. The data is randomly split into 2 files of equal size. The first file is used as training data for a neural network. The neural network "learns" the relationship between the shop factors and the correct number of kanbans needed from the training data. After the training phase, the neural network is tested on its "associative" ability to determine

how well it predicts the correct number of kanbans for the shop scenarios in the second file (data it has never seen). Results are given for different network paradigms to determine the best paradigm for predicting the number of kanbans in a dynamic JIT shop. The neural network is also used as a tool for factor screening. Each factor is analyzed to determine its relative importance in kanban prediction. Statistical tests are used to gauge the importance of the dynamic information as well as to examine the relevance of various factor groupings. The results have practical implications for firms that have adopted, or are considering, the JIT technique.

Dedication

To my parents, Jesse A. and Sylbia S. Wray.

Acknowledgements

I would like to thank my committee chair Dr. Terry R. Rakes for his guidance, support, patience, and inspiration. His insightful comments and advice throughout my dissertation were graciously appreciated. I am especially grateful for the time he set aside during the final stages of my dissertation. I would also like to convey my appreciation for his assistance and guidance in my job search process.

I would like to thank Dr. Loren P. Rees for his continued support throughout my graduate studies at Virginia Tech. Loren's friendship and encouragement were pivotal for me during my years in Blacksburg.

I would also like to thank Dr. Edward R. Clayton, Dr. Roberta S. Russell, and Dr. Robert T. Sumichrast for the time and effort they spent as members of my dissertation committee.

I would like to thank Dr. Laurence J. Moore for inspiring me to pursue a Ph.D. in Management Science.

I would like to thank Dr. Bernard W. Taylor for his leadership and financial support during my graduate studies.

I would like to thank all the graduate students in the department of Management Science for their support and friendship during my time at Virginia Tech.

I would like to thank the faculty of the Department of Production and Decision Sciences at UNCW for their assistance and support while I finished my dissertation.

Finally, I would like to thank Gerry, Sylvia, Tracy, and Teena for their excellent secretarial support while I was at VPI.

Table of Contents

List of Illustrations

Chapter 1

INTRODUCTION

The success of the Japanese just-in-time (JIT) with kanban inventory control technique has motivated many manufacturing firms world-wide to implement similar systems to remain competitive. In order for a JIT system to be cost effective, certain conditions such as smooth and stable demand, low setup times, small lot sizes, and highly flexible workers need to be met (Huang et al. 1983, Krajewski et al. 1987). For a more complete listing of requirements, see Lee and Ebrahimpour (1984). Some Japanese firms have developed an internal and external production environment allowing them to implement JIT with its many advantages including low inventory levels, minimum defects, high quality and respect for humanity. These factors contributing to the success of these firms have drawn a great deal of attention from manufacturing competitors around the world. Consequently, JIT with kanban is gaining popularity in various manufacturing environments. A critical question that must be answered is how to implement, control, and sustain JIT in various environments.

The identification of critical shop factors to control when implementing JIT is an important issue in the development of a highly productive manufacturing system. When critical factors have been identified, a control mechanism is needed to adjust the number of kanbans (inventory level) at each workcenter within the shop. Firms wishing to implement JIT must develop a technique to dynamically control the number of kanbans circulating based on both endogenous variables (shop conditions) and exogenous variables (such as product demand and supplier schedules).

The Japanese determine how many kanbans to use based on trial and error or simple heuristics developed over many years of implementation. However, predicting and controlling the number of kanbans in a dynamic environment where critical factors are changing over time is a complex decision involving many stochastic factors. A methodology that allows a firm to exercise control over shop efficiency by identifying critical factors and to "look ahead" and predict the number of kanbans needed at a workcenter would be a very valuable tool, negating the long process of trial and error.

This research investigates using the artificial intelligence (AI) technique of neural networks as a tool for factor screening and functional synthesis in an unstable JIT production setting. A neural network "learns" the relationship between shop factors and the optimal number of kanbans needed for efficient shop operation. The neural network predicts the number of kanbans based on static factors found in the shop as well as "dynamic" factors inherent in stochastic production environments. The explanatory power of a neural network is used to determine the static and dynamic variables that are critical when unstable shop conditions exist. The ability to predict and control the correct number of kanbans is a very valuable tool for any company adopting the Japanese JIT technique.

A simulation model incorporating less than ideal conditions is used to replicate a stochastic, highly dynamic JIT shop. The simulation model replicates various shop conditions (a combination of factor levels) and the relative performance of different numbers of kanbans at a workcenter. The optimal number of kanbans for each factor level is determined by comparing a total cost function for each kanban level. Only the data for the lowest cost number of kanbans is used as training data for the neural network. The costs for all levels of kanbans are used to evaluate shop performance for a given number of kanbans. A neural network is developed that uses the shop simulation output as training data. After the neural network is trained it can predict the optimal number of kanbans based on any combination of factor levels. The neural network is tested on different shop scenarios to determine its effectiveness in predicting the correct number of kanbans for the workcenter.

Different issues on neural network development are analyzed to determine the best neural network model to use for prediction and control problems in a JIT production setting. Different network paradigms are evaluated to provide an overall perspective of the effectiveness of different solutions to the problem. Network construction issues such as what type of learning schedule should be used or how many processing elements should be included in the hidden layer are presented. A "refined" methodology is given along with results of this new system.

Problem Statement

The JIT with kanban production system is not just a method of controlling material flow during production, it is a philosophy that must be embraced by all aspects of a firm in order for the system to be successful. The economic, philosophical, and cultural conditions found in Japan provide an environment that is conducive to the success of the JIT system. If these conditions are not met, the implementation of JIT may be hampered. The JIT total production system demands preparations in the production system including scheduling a smoothed sequence of products at the final assembly, machine layout designs, standardizing operations, shortening setup times, etc. (Monden 1983). Such factors as lifetime employment and a frozen master production schedule are typically unattainable by most manufacturing firms. However, vendor-producer cooperation, well- and cross-trained workers and reduction of set-up times are factors that can be addressed. Two problems faced by firms implementing JIT in "non-optimal" situations are addressed in this research. The first problem analyzed is how can the optimal number of kanbans be predicted? The second problem is the determination of factors, or groups of factors, critical to JIT implementation.

Typically in the Japanese environment the number of kanbans is determined from the average demand, production leadtime, and a safety factor (Monden 1983). The level of inventory is controlled by reducing or increasing the number of kanbans and/or container size. This kanban adjustment process is a long trial and error procedure that may be an obstacle for implementation of JIT in firms unable to endure a long transformation from one system to another. Therefore, an alternate method for predicting and controlling the number of kanbans must be developed.

The methodology is extended to determine critical shop factors that need to be controlled in order for the system to run efficiently. After important factors are identified, they must be monitored to determine when and by how much the number of kanbans in the system must be adjusted. The decision is complicated by such factors as lumpy demand (seasonal fluctuations), varying production leadtimes, and other dynamic factors affecting the shop.

Purpose of the Research

The purpose of this research is to develop a refined methodology that can be used by any firm wishing to apply the JIT with kanban technique. The methodology allows a firm to determine which shop factors are important in controlling the shop, and based on these factors, to predict the number of kanbans needed in the production environment the firm must operate in.

The artificial intelligence technique of neural networks is the tool used to accomplish this goal. A neural network is chosen because of its ease of use, ability to identify factors important in shop performance, and its touted forecasting ability. Existing shop data, if available, or data generated by a simulation model can be used to train the network and to evaluate its performance for a specific shop.

Scope and Limitations

This research focuses on the problems facing firms employing the JIT with kanban technique in a stochastic production environment. More specifically, a methodology is developed that can be used by an individual firm to deal with their unique environment. A method of determining which factors, specific to a given shop, are critical for JIT to be successful is presented. A model is developed to "learn" important relationships in the JIT environment and to predict the optimal number of kanbans at a workcenter based on the interaction of dynamic shop factors. The tool used in this research is a neural network. Neural networks are currently being used in many applications including language processing, data compression, character recognition, combinatorial problems, pattern recognition in images, signal processing, financial and economic modeling, servo control, and functional synthesis. This research focuses on using neural networks to synthesize the surface of the function driving shop performance. After the methodology is developed using standard techniques it is refined by examining different aspects of the neural network to determine the most effective model to accomplish the desired goals.

The manufacturing process chosen for this research is a multi-line, multi-stage, repetitive assembly shop since this type of environment is the most typical for the application of JIT. A JIT pull-system could be applied in various manufacturing systems such as group technology, cellular manufacturing, flexible manufacturing, or a job shop. A limitation of this work is that the results and conclusions may not be generalizable to all of these manufacturing systems. Further research into each system would be needed to determine the applicability of the methodology for that particular system.

This research does not address the process of collecting historical shop data to use for network training. If past data are available, a judgement must be made as to the effectiveness of the number of kanbans used for a given period with a set of inputs. For instance, if 3 kanbans were used for a certain combination of factors the manager must determine the efficiency of the number of kanbans used. If shop performance is considered poor when 3 kanbans were used, this data cannot be used since it would train the network to perform poorly.

If historical data are not available a simulation model must be built to generate usable input for network training. The model must be validated and continuously updated to provide usable training data for the neural network. The construction of a simulation model could be very costly and time consuming for a company, thus reducing the benefits gained by using a neural network.

Plan of Presentation

Chapter 2, entitled Literature Review, is a summary of research papers presented and published relating to Just-In-Time with kanban production and the use of neural networks to solve factor screening and functional synthesis problems. This chapter provides the background for the analysis presented in subsequent chapters.

Chapter 3, entitled Methodology Overview, gives a detailed account of the steps involved in the research. The purpose and description of each step provides the reader with a comprehensive listing of the processes used in the research. A listing of factors examined in the study is also included in this chapter.

Chapter 4, entitled Dynamically Adjusting the Number of Kanbans in a Stochastic Just-In-Time Environment, presents the neural network solution process. The data generation process is explained to give the reader a better understanding of what was involved in generating data to replicate an actual JIT shop. The neural network model development, training, and recall processes are explained. The issues surrounding performance evaluation of the neural network are discussed and a final evaluation method is determined. Results of the methodology are presented and analyzed. The last section of chapter 4, entitled Network Construction Enhancements to Improve Network Performance, identifies how neural network construction and application can be improved to enhance its performance. Data on improved network performance is tabulated and implications of these improvements are given.

Chapter 5, entitled A Neural Network Approach to Identify Critical Shop Factors in a Stochastic JIT Environment, presents an evaluation of each factor included in the study to determine which factors are "critical" in the solution process. A factor screening process is used to individually test the significance of each factor as well as select groups of factors.

Chapter 6, entitled Conclusions, evaluates the research and looks at possibilities of implementing the proposed system and future research in the area.

Chapter 2

Review of Related Literature

The following sections briefly review past and current research dealing with determining critical shop factors in a JIT with kanban environment and predicting the number of kanbans needed at a workcenter in the shop. A brief history of the artificial intelligence technique of neural networks and some current applications are also presented. Although research is available in both areas, very few studies have dealt with the analysis of dynamic shop factors critical to the effective application of JIT and determining a methodology to dynamically predict the optimal number of kanbans. This work appears to be the first to apply neural networks to these specific problems.

Just-In-Time with Kanban

The success of the Japanese Just-In-Time with kanban technique has prompted many researchers to study this system. A complete description of Toyota's application of JIT is provided by Monden (1983). Krajewski et al. (1987) simulated a JIT system to determine if the Japanese approach could perform weil in the manufacturing environments found in the United States. Their research also assessed which factors in a production environment have the greatest impact on performance - regardless of the system used. The kanban system was compared to an MRP system using simulation. A detailed questionnaire was sent to a panel of managers to form a list of factors they felt were important to the performance of their respective manufacturing environment. The impact of each factor was studied by means of many simulation runs. The individual factors found to be the most important were: inventory, process, buffer mechanism, and customer influence. Other factors worthy of consideration were: product structure, facility design, and vendor influence. The authors found that the kanban system, by itself, is not the key to improving shop performance. It is only one part of an overall manufacturing philosophy that must be adopted to reduce inventory, increase productivity, and improve customer performance. It was shown, however, that the improvement of certain factors are more likely to have a greater payoff than others.

The impact of stochastic processes on a JIT system has been studied by means of simulation. Huang et al. (1983) provided a simulation model to determine the adaptability of JIT to a multiline, multistage production system with variable processing times, variable demand rates, variable master production schedule, and imbalances between production stages. They found that variable processing times and demand rates have a definite impact on the mean and variance of overtime and production output. If was also found that additional kanbans do not alleviate regularly occurring bottlenecks.

They also concluded that any company considering changing to a kanban system would need a lengthy transition period. They estimated a transition period of at least one year for companies with the variable operating conditions tested. It was recommended that companies wishing to implement JIT institute a great deal of worker training and cross-training to standardize machine processing times and setup times. Some methods of instigating worker loyalty were also suggested for enhancing the employee-employer relationship necessary to retain workers after the extensive training process. Their final conclusion was that companies with operating conditions similar to those found in Japanese companies that have successfully applied JIT have the best chance of successfully implementing a JIT system.

The identification of factors critical to the implementation of JIT with kanban in a production environment is a very important preliminary step in determining the feasibility and eventual performance of the system. Philipoom et al. (1987) identified "static" factors influencing the number of kanbans by an analytical analysis of the equation for determining the number of kanbans given by Monden (1983). The equation:

$$\text{Number of Kanbans} >= \frac{(\text{demand})(\text{lead time})(1+\text{safety factor})}{\text{container capacity}}$$

is manipulated to derive the following equation:

$$Z_p = (v^{1/2}/CV)[(1/util) - 1], \quad 0 < util <= 1, \quad CV <> 0$$

where Z_p is the number of standard deviations that the cycle processing time must be greater than its mean in order for container demand cycle to equal cycle processing time.

This equation is analyzed to identify the factors that influence the probability of a backorder, and thus the need for kanbans. The factors identified are:

v - Throughput velocity (i.e. the rate at which items flow through a workcenter machine)

CV - Coefficient of Variation of the processing time at the workcenter.

util - Utilization of machines at the workcenter

Autocorrelation of machine processing times - violation of the independent and identically distributed assumption in the derivation of the equation

Each factor within the equation was analyzed as to how it would impact on an American production manager wishing to implement a JIT with kanban system. They found that the number of kanbans needed would increase if any of the following occurred: throughput velocity decreased, the CV of processing increased, machine utilization increased, or machine processing times were autocorrelated.

This research also described an approach for determining the number of kanbans to use at a workcenter based on product demand and workcenter lead times. A formulation basically the same as that used by the Toyota Company was used to determine the minimum number of kanbans at a workcenter. The formula is:

$$n >= D_m L_{.95}$$

where n is the number of sets of kanbans for an item at a workcenter, D_m is the maximum demand for that item's final product in containers; and $L_{.95}$ is the maximum lead time at that workcenter (determined by the 95th lead time value from 100 simulated lead times ranked in order of magnitude from lowest to highest).

The approach was based on modeling each workcenter separately as a queueing system in order to generate the lead times via simulation.

The methodology was used to set the initial number of kanbans at a workcenter for an example production system. The system was simulated to show that the initial number of kanbans selected did meet the criteria of satisfying all demand with no backorders. The same simulation model was used to show the impact of the various factors identified earlier in the paper. The simulation results confirmed the influential effect of each factor.

Gupta and Gupta (1989) used a dynamic simulation model to demonstrate how endogenous shop factors such as work-in-process (WIP) inventory, capacity utilization, and final product shortages affect the behavior of a JIT with kanban system. They simulated a JIT with kanban production flow shop to determine the impact of different management policies dealing with changing the number of kanbans or the size of a container, machine breakdowns, and processing time uncertainties.

The authors found that the combination of container size and the number of kanbans can be critical to system efficiency. It was found that increasing the size of containers and decreasing the number of kanbans leads to a higher level of WIP inventory. Also, the effect of a production stoppage (machine breakdowns) in a given stage demonstrated the feedback nature of a kanban system. Production in other stages decreased or stopped to ensure that excess inventories were not produced.

Increased variability in processing was found to decrease the production rate and increase shortages. When this occurred an increase in the number of kanbans (and thus an increase in inventory) was necessary in order to meet demand.

Two other interesting conclusions from the simulation study were the importance of vendor reliability and the balance of all stages of the kanban system. It was shown that simply increasing the number of production kanbans at a workcenter would not increase the production rate when other parameters remain constant. An important conclusion of this research is the extreme importance placed on determining the correct number of kanbans in a JIT system.

Deleersnyder et al. (1989) studied the operational control facet of implementing a kanban controlled JIT system. They developed a discrete time Markov model of a N-stage single-item manufacturing process with N machining cells in a series. The shop model included capacity constraints, stochastic machine reliability and demand variability as the factors affecting the performance of the shop. The production rates, inventory levels, and product demand are used to derive a one-step transition probability, Pr[I(n-1),I(n)], that describes all possible transitions within the shop.

In order to analyze the system, several key performance measures were needed. The authors tracked the average amount of total inventory (to measure operating costs), the average and variance of backlogs (to measure service level), percentage of lost demand, and average job flow time. As an illustration of the model, a 3-stage serial production system was evaluated. This model required 20 sec. to 15 minutes of CPU time on a Siemens 7550 mainframe computer. The authors comment on the fact that a model of 5 or more stages (a more realistic case) would require significantly more CPU time and computer memory. The effects of the number of kanbans, machine reliability, demand variability and safety stock requirements on the performance of a kanban controlled pull system was given. The most significant conclusion is that the model demonstrated how incremental improvement can be achieved by iteratively lowering the number of kanbans and increasing machine reliability. It was also shown that the downstream kanban loop is the key in improving multistage system performance given the system parameters studied.

Chaudhury and Whinston (1990) presented a decentralized, adaptive control methodology for flow shops. The methodology is based on machine intelligence techniques and is primarily designed to make kanban-type systems more flexible and adaptive without disturbing the control simplicity of the kanban system.

Research on determining the number of kanbans in a deterministic shop (constant demand, constant processing times, etc.) is available. Bitran and Chang (1987) developed a mathematical programming model for a deterministic multi-stage capacitated assembly-tree-structure production system. The model provides a mathematical programming formulation of a JIT with kanban system to assist managers in determining the number of circulating kanbans at each stage of the production process. The initial model, which is nonlinear, is transformed into an integer linear model to provide a more tractable solution process that provides the same optimal solution as the nonlinear model in terms of the decision variables controlled by managers. Three special cases of interest are constructed on the basis of container size between stages.

The model development process in an actual production shop would require a very skilled mathematical programmer to formulate the model for that particular shop. The model would be very situation specific and require a reformulation if any of the static factors used in developing the model changed. None of the models suggested in the research allow for any type of uncertainties such as demand or machine breakdowns. If a workable model is formulated, the solution process for the integer and general linear programming models presented would require a significant amount of computer time due to the size and complexity of the models.

Philipoom et al. (1990) developed a heuristic approach to simultaneously specify the container size, number of kanbans, and sequence of jobs in a multiple-level, capacitated JIT shop. The solution process presented in the research is called JACKS (JIT Algorithm for Containers, Kanbans and Sequence). The JACKS procedure is based on an algorithm called MOPS (Method Of Prime Subperiods) developed by the same authors to solve the economic lot scheduling problem. MOPS is used to determine container sizes and product sequence at the final stage of the production process, then JACKS determines container sizes, product sequence and the number of kanbans at previous stages.

The JACKS algorithm uses as input the final stage container sizes and product sequence determined by MOPS. The heuristic assumes a "static" or deterministic shop with the following shop parameters assumed known and constant for all workcenters and/or products:

- I_j = the set of products produced at workcenter j
- D_i = the demand for product i per unit time
- S_{ij} = the setup time for product i at workcenter j
- P_{ij} = the processing time for product i at workcenter j
- C^s_{ij} = the setup cost for product i at workcenter j
- C^h_{ij} = the holding cost for product i at workcenter j

The JACKS algorithm is described by the following 6 steps:

1. Initialize
2. Load the final workcenter
3. Transmit the schedule to all other workcenters
4. Check feasibility
5. Examine Cost
6. Iterate

An example shop with six workcenters producing 5 similar products is used to illustrate the JACKS procedure. The JACKS procedure was shown to be computationally feasible, however since there is no other approach for simultaneously determining the container size, product sequencing and the number of kanbans in a JIT system, the solution cannot be compared to other similar algorithms.

The authors point out that the computational time of the JACKS heuristic does not increase exponentially with an increase in the number of workcenters; however it does increase exponentially with the number of products.

Different methods of adjusting the number of kanbans in a stochastic setting have been presented. Rees et al. (1987) developed an eight-step procedure using estimated values of leadtime to dynamically adjust the number of kanbans in a JIT production system. The procedure is based on the equation:

$$n = [\text{Demand} * \text{Leadtime} (1 + \text{safety factor})]$$

used by the Toyota Motor Company [Monden, 1983]. The number of kanbans at a workcenter is adjusted periodically based on a forecast of demand and collected observations of leadtime at the workcenter during recent periods. The procedure proposed by Rees et. al. was developed for firms that do not exhibit the conditions found at Toyota and therefore cannot use the Toyota equation directly. In these firms, the demand and leadtime for a product are dynamic and therefore must be estimated. The number of kanbans at a workcenter is adjusted periodically based on a forecast of demand and collected observations of leadtime at the workcenter during recent periods. The eight steps of the methodology are:

1. Startup - If major changes have been made to the shop, the changes should be permitted to die out
2. Measuring Period 1 - Obtain autocorrelation estimates of the container leadtime series so that statistically independent observations can be observed
3. Measuring Period 2 - Use the independent observations from measuring period 1 to estimate the density function of leadtimes for an item at a workcenter
4. Forecast Demand - A forecast of the next period's demand is made using standard company methods

5. Determine the PMF for the Number of Kanbans - Estimate a probability mass function for n (number of kanbans) given the estimated density function of leadtime and the demand forecast
6. Determine the Minimum-Cost Number of Kanbans - Use the PMF for n, the holding cost of an item, and shortage cost for unmet demand to determine the minimum-cost number of kanbans
7. Action Step - Adjust the number of kanbans at a workcenter
8. Settle Period - Ensure that the workcenter has sufficient time to settle down and adjust.

The methodology is demonstrated by three example JIT shops. The first shop is simulated to illustrate each of the eight steps. A second example shop is used to show how well the methodology adjusts to a one-time misspecification in the number of kanbans. The purpose of the third example shop is to study the cost effects of training workers. Although the eight-step procedure does provide a safe method for kanban adjustment, it is mathematically intensive. Each loop through the system requires a great deal of data collection and statistical manipulation.

Groenevelt and Karmarkar (1988) described a case study for a batch flow shop that adjusts the number of kanbans by explicitly forecasting the average level and variability of demand. The production process studied differed from the Toyota system in the following ways:

- seasonal demand
- forecast release of kanbans (as opposed to a reactive release)
- lot splitting or consolidation
- contingency allowance for kanban adjustment
- "disassembly" processes

The forecast release of kanbans added a "push" element to the typically "pull" kanban approach. The number of kanbans are adjusted as a function of the average level and variability of demand.

The approach was implemented by the Schlegel Corporation. The system enabled management to evaluate the backlog load on the manufacturing floor by looking at the number of kanbans in the system. This enabled them to more efficiently schedule production based on improved information, and the forecast for demand.

The positive impact of applying JIT in several U.S. manufacturing companies is presented in an article by Waters (1984). Omark Industries saved an estimated $7 million in inventory carrying costs with its ZIPS (Zero Inventory Production System) version of JIT. T.D. Shea Manufacturing Inc., of Troy, Mich., a producer of plastic products for the automotive industry, developed a Nick-In-Time system. The company incurred some initial start up costs but expects to save a significant amount in reduced inventories. Other companies that have adopted a "JIT-like" production system include: the Harley-Davidson Motor Company, the big four U.S. automakers, Hewlett-Packard, Motorola, Westinghouse Electric, General Electric, John Deere, and Black & Decker.

Neural Networks

Historical Background

Artificial intelligence and neural computing were introduced by Marvin Minsky, John McCarthy, Nathanial Rochester, and Claude Shannon at a conference in 1956. During the conference, Rochester presented a neural network model consisting of several hundred simulated neurons and interconnections. In 1959 Bernard Widrow developed adaptive linear elements called Adaline and Madaline for applications in speech recognition, character recognition, weather prediction, and adaptive control.

The first major research project in neural computing was the development of an element called a Perceptron by Frank Rosenblatt in 1957. Rosenblatt's work inspired additional research in neural computing until Marvin Minsky and Seymour Papert published a critique of the Perceptron in their book **Perceptrons** (1969). Minsky and Papert concluded that the Perceptron and neural computing were basically not "interesting" topics. **Perceptrons** greatly decreased the amount of funding available for further research in neural computing. Despite Minsky and Papert's book several researchers such as James Anderson, Teuvo Kohonen, and Stephen Grossberg continued work in neural computing. Renewed interest in neural computing was generated when John Hopfield presented a paper on neural computing to the National Academy of Sciences (1982). Hopfield's outstanding reputation combined with his personal charisma provided the impetus for a rejuvenation in neural network research.

Current Research in Neural Computing

Neural networks have been used in a variety of applications. In the area of language processing, Sejnowski and Rosenberg (1987) used neural networks to convert text to phonetic representations, which were ultimately converted to speech. A neural network for character recognition was developed by Burr (1987). This network can recognize handwritten characters and convert them to computer input. In the field of finance and economics, a neural network was shown to be more effective than standard forecasting techniques in predicting the S&P 500 closing stock price at the end of each week.

The interpolative nature of neural networks makes them suitable for functional synthesis. A neural network has been developed to determine the impact point of a projectile fired from a cannon (Klimasauskas et al. 1989). The network can synthesize the complex multidimensional surface representing all possible impact points and approximate the function generating the surface. Neural networks are being used to perform automated factory inspection, process control with design enhancement, operations analysis (Kinoshita 1988), and due-date prediction (Philipoom et al., 1990). Neural networks have also been applied to the areas of image compression, combinatorial problems, pattern recognition, and signal processing.

Current Research in JIT Using Artificial Intelligence

Current research involving the application of neural networks to JIT production-type problems is limited. Rees et al. in a working paper (1987) suggest an artificial intelligence (AI) approach incorporating machine learning to address the number of kanbans problem.

Rees, et. al. (1987) propose an artificial intelligence approach using machine learning to examine shop conditions to develop heuristics for determining the number of kanbans. An automatic mechanism that will adjust predefined rules and create new rules or heuristics is the future goal of the research. An expert system based on the knowledge of how leadtime affects the number of kanbans could be developed using an expert system shell. The process of "learning" would involve a merging of simulation output or actual data with a program using this information to generate new rules for the expert system. The difficulty of developing the system is deciding on how to "learn" from the given information. Machine learning schemes such as parameter adjustment or signature tables suggested by the authors would require an "expert" programmer that could implement a strategy via an integrated system. Such a system would require extensive time and effort to program the learning scheme and has yet to be built.

Rakes et al. (1990) developed a methodology using the AI technique of neural networks to determine the optimal number of kanbans based on static shop factors. A multi-line, multistage JIT production system was simulated to determine the optimal number of kanbans for a given combination of variable product demand, variable setup time, and different ratios of backorder to holding costs. The data generated were used to train a neural network that accepts, as input, the operating conditions of the shop (coefficient of variation of demand and setup, backorder/holding cost ratio), and then outputs the number of kanbans to use for the next period. The trained neural network predicted the number of kanbans that produced a total cost between

6.72% and 10.23% from optimal. The purpose of this research was to show the feasibility of this approach. This current research is an outgrowth of that idea. The motive for this work is to provide an easy to use and efficient neural network based methodology to enhance the performance of JIT in firms facing a dynamic production environment. The system will provide a quick and simple method to determine critical shop factors and the required number of kanbans for a given workcenter in a truly "dynamic" shop, considering as input both endogenous shop conditions and exogenous factors (such as demand variability, machine utilization, supplier schedules, etc.).

Chapter 3

Methodology Overview

The steps followed in this research are (see Figure 1 for a flowchart describing the sequence of steps):

1. Generate a Preliminary List of Variables to Observe During Shop Simulation

2. Develop a Simulation Model of a JIT Shop to Generate Training Data

3. Develop and Train a Neural Network

4. Predict with the Network

5. Evaluate the Performance of the Neural Network

6. If the Network Performance is Satisfactory, Continue; Else Develop and Train a New Neural Network Model and Return to Step 4

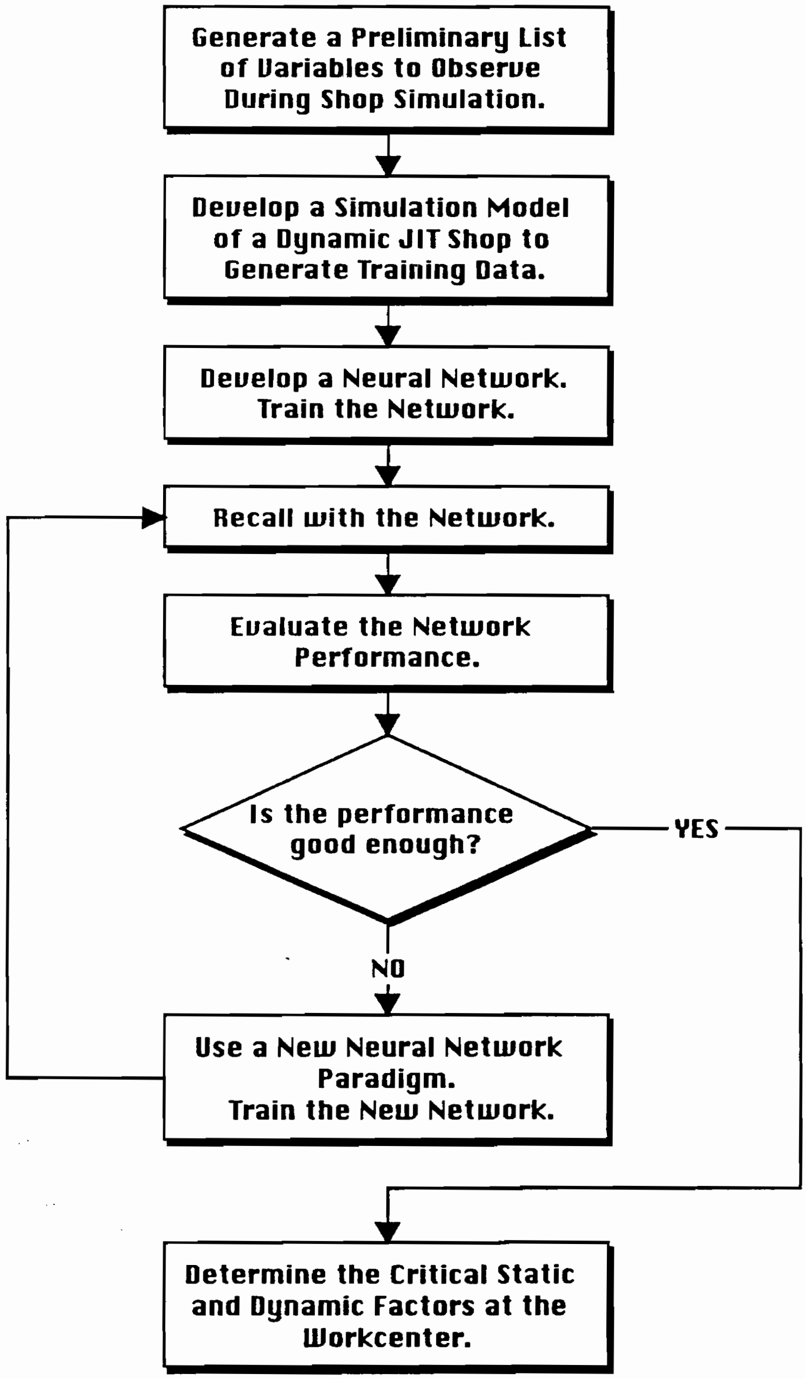


Figure 1. Methodology Overview

7. Determine the Critical Static and Dynamic Factors in the Shop

The final product of this research will be a refined methodology using a neural network for determining "critical" shop factors and predicting the optimal number of kanbans in a highly dynamic production environment. The following sections discuss the appropriateness and sequence of each step.

Identification of Important Factors

An important question facing firms evaluating the feasibility of implementing JIT is the applicability of this production system to their individual environment. A great deal of research has been undertaken to determine critical factors such as multi-functional workers, reduction of setup times, and frozen demand schedules in the Japanese Toyota production environment (Monden 1983).

Since many of the conditions found in Japan cannot be duplicated, a modified JIT system must be used. Relevant factors critical to efficient shop operation need to be monitored and controlled. A method of determining which factors are important is a very valuable tool for a shop considering or already using JIT. This research will use neural networks to perform "factor screening" on its inputs to determine which factors have a high influence on network predictive ability. Initially, many shop factors will be used as input to a neural network. Some input factors will be results from the previous period's operations while other factors will be past and current operating conditions for the shop. All factors will be used to determine the number of kanbans to use during the present period. If a parameter of a probabilistic factor, such as demand, remains unchanged over the two periods considered the factor is considered "static." If, however, a parameter such as the mean or standard deviation of demand changes between periods the input factor is considered "dynamic." The dynamic case requires a great deal of computational effort but is the more robust case since it can replicate a "changing" production environment found in many firms. Factors chosen to include in the study were determined by a review of JIT literature, analysis of a simulation of a "static" shop, and an analysis of all factors that logically seemed influential in a "dynamically" changing shop.

Based on the previous work of Philipoom et al. (1987), Gupta and Gupta (1989), Deleersnyder et al. (1989), Ragatz and Mabert (1984), and Rakes et al. (1990), the initial factors chosen for study are as follows:

Input Factors

Shop conditions at the Target Workcenter
for the Present and Previous Periods:

1. Demand Variability
2. Machine Processing and Setup Variability
3. Supplier Delivery Variability

Resultant Factors

Shop Performance Criteria From the Previous
Period for the Target Workcenter

1. Leadtime - Time Between When a Production Kanban is Received and the Item is Produced
2. Finished Goods Inventory - Completed Production at the Workcenter
3. Work-In-Process Inventory - Incomplete Production Within the Workcenter
4. Overtime Needed - Amount of Time Over One Shift Needed to Meet Demand

5. Kanban Circulation Rate - Number of Times a Kanban Circulates Within the Workcenter Per Time Period

6. Withdrawal Kanban Waiting Time - The Time a Withdrawal Kanban Waits at the Withdrawal Post

Each factor will be examined over many different shop scenarios. A contribution of this research will be to demonstrate that a by-product of a neural network approach is the ability to perform factor screening on the inputs to the network.

Generating Dynamic Shop Data Using Simulation

The approach employed in this research is to apply the artificial intelligence technique of neural networks to prescribe dynamically (through time) the number of kanbans in a JIT shop based on endogenous shop conditions and exogenous variables. Therefore, input and output data from a JIT shop is needed for training and evaluating the neural network. These data are shop specific; moreover, the shop conditions under which the training data are obtained should emulate the shop environment as closely as possible. Changing environments must be used to truly represent the daily stochastic sub-optimal (for JIT) conditions faced by many firms implementing JIT.

For this research, simulated data will be used rather than "real" data. Simulation is chosen for three reasons. First, once the simulation model is built it is fairly easy to generate large amounts of data. Second, actual shop data are difficult to obtain and may contain errors. Third, a simulation model has the ability to generate data under extreme conditions that may be unavailable from actual shop data.

The shop chosen for this study is an extension of the shop used by Rakes et al. (1990). Their model was based on the multi-line, multi-stage kanban process illustrated by Huang et al. (1983). This basic model is expanded to allow for multiple workcenters, multiple products, setup time, and collection of shop statistics over two time periods. The model consists of six workcenters with two final products. The particular workcenter for which we wish to estimate the number of kanbans is workcenter one in Figure 2. By examining the product routings, it is evident workcenter one processes parts coming from five different workcenters (2-4, 5,6) to produce two final products. The processing at workcenter 1 is deliberately made "complicated" with the inclusion of waiting for various components, setups, etc., to make the investigation more realistic.

The shop is simulated for two days of operation. The standard deviation of either product demand, machine processing, or vendor supply is changing over the two day period to replicate the dynamic environment.

SLAM II (Pritsker 1986) is used as the simulation language to model the shop. A portion of the Slam II model is shown in Figure 3. Since the basic purpose of this research is not the development of a JIT shop model, and since the model so closely follows the approach taken in Huang et al. (1983), further discussion of SLAM II modeling details is not included.

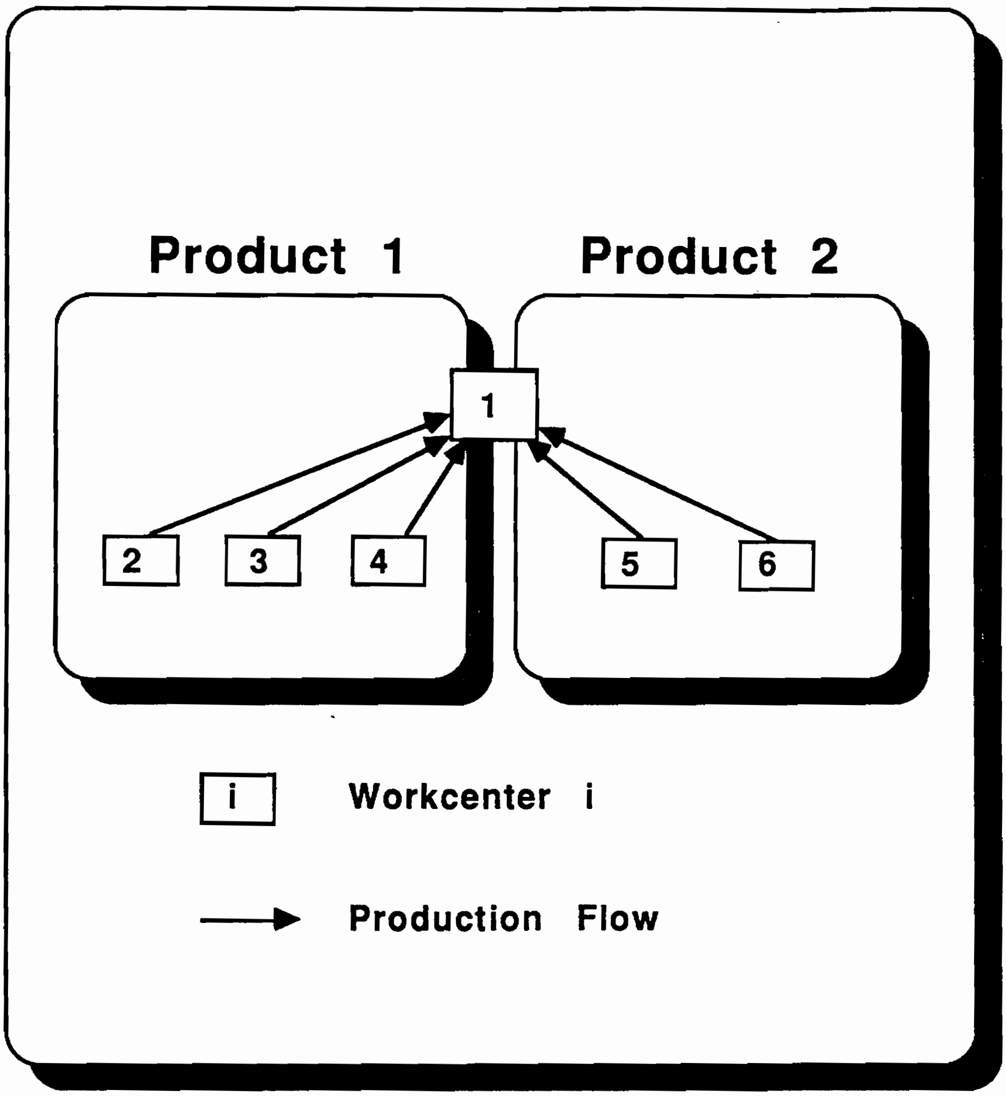


Figure 2. JIT Shop Production Flow

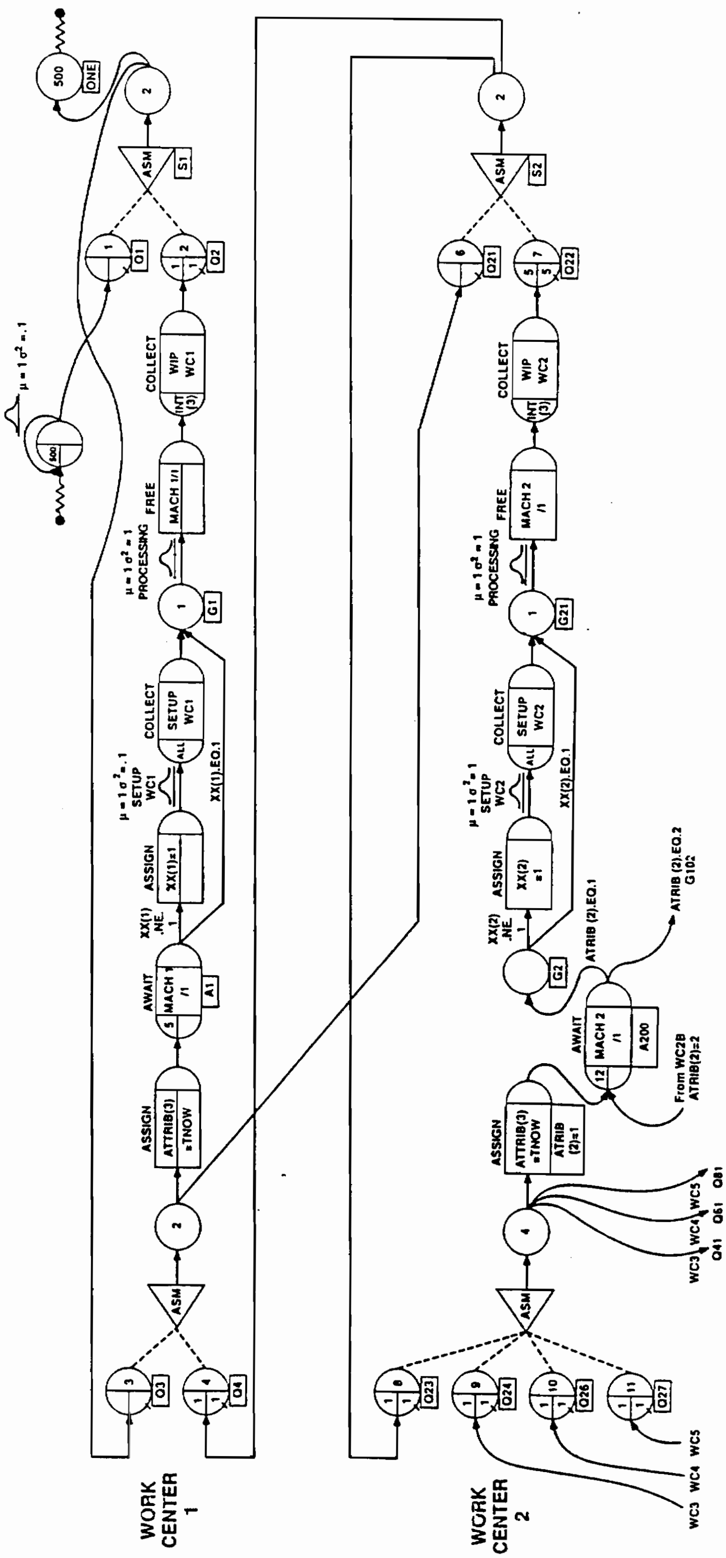


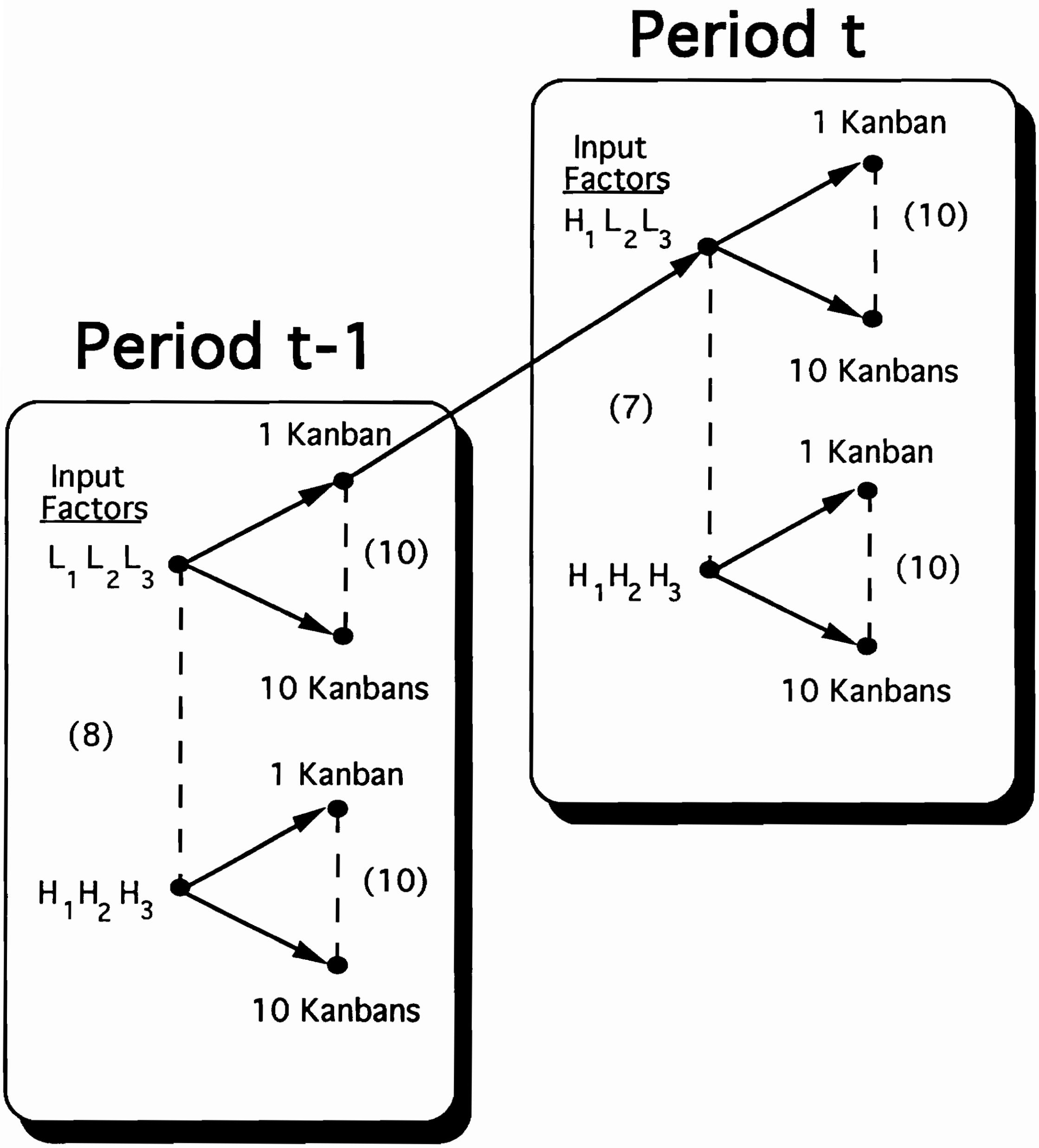
Figure 3. Slam II Network for the JIT Shop

Experimental Design

Since actual shop data are unavailable for this research, a sample data set must be generated via simulation to use as training data for the neural network. The data set consists of shop factors over a two period time frame and the number of kanbans used in each period. A graphical representation of the experimental design is provided in Figure 4. A multi-period design demonstrates the ability of the neural network to "learn" how past and present shop conditions impact on shop control decisions in a "dynamic" production environment. The experimental design used to generate the data set takes into consideration all factors at all levels that are included in this study. Every combination of input factors in period t-1, possible number of kanbans in period t-1, combination of input factors in period t, and possible number of kanbans for period t is represented in the data set. Therefore, every conceivable shop scenario and the optimal number of kanbans for that scenario, is included in the simulation data set. By including all factors at all levels, the scope of the data used for training and testing the neural network will be vast enough to allow detection of influential factors.

In order to describe all possible shop scenarios over two periods, the dynamic factors included in each period are:

1. Demand Variability
2. Machine Processing Variability
3. Supplier Delivery Variability



L_i = Low Coefficient of Variation for Variable i

H_i = High Coefficient of Variation for Variable i

Figure 4. Experimental Design

These three shop factors can be controlled by initially setting the parameters of their probability distribution at the beginning of each simulation run. These factors will be called "controllable" or "input" factors. The coefficient of variation (CV) of each input factor will be varied over two different levels - high and low. In order to determine the dynamic effect of changing distribution parameters, the three controllable input factors will be varied over all possible combinations of high and low. The scheme used to represent an input factor and its level in Figure 4 is:

L_i = Low Coefficient of Variation for Variable i

H_i = High Coefficient of Variation for Variable i

NOTE: The subscript i corresponds to the number of the input factor in the list above

The following branch:

$L_1L_2L_3 \rightarrow$ 1 Kanban $\rightarrow H_1L_2L_3 \rightarrow$ 1 Kanban

represents the shop scenario of all factor CV's low in period t-1 with 1 kanban, while in the next period, period t, the CV of demand changes to high while the CV's of processing and vendor supply are held at low with 1 kanban.

All other factors used in the training of the neural network will be measures of shop performance from the previous period. These factors are "resultant" or "output" since they are shop performance measures resulting from the interaction of the input factors and the number of kanbans from the previous period. The six resultant factors that will be tracked during each simulation run and included in the data set are:

1. Leadtime - Time Between When a Production Kanban is Received and the Item is Produced
2. Finished Goods Inventory - Completed Production at the Workcenter
3. Work-In-Process Inventory - Incomplete Production Within the Workcenter
4. Overtime Needed - Amount of Time Over One Shift Needed to Meet Demand
5. Kanban Circulation Rate - Number of Times a Kanban Circulates Within the Workcenter Per Time Period
6. Withdrawal Kanban Waiting Time - The Time a Withdrawal Kanban Waits at the Withdrawal Post

In the previous period a full-factorial design is used with all three factors at each level - High and Low. Therefore the number of period t-1 factor level combinations is: $2^3 = 8$. For each combination of period t-1 input factors the number of kanbans must be considered. Because there must be at least 1 kanban, the design starts with 1 kanban for each input factor combination and then adds an additional kanban until the number of kanbans reaches 10. A maximum of 10 should be sufficient given the interactions of the workcenters within the shop. The range of 1 to 10 kanbans is not a limiting factor since the data set consists of only "what-if" data. In an actual shop the range of the number of kanbans would be based on the particular shop's variability.

For each period t-1 combination there will be $2^3 - 1 = 7$ period t factor combinations (since a duplication of a period t-1 combination for period t would represent an "unchanged" shop which is not relevant for this study). Therefore, the design is up to 8 (period t-1 input combinations) X 10 (period t-1 kanbans, from 1-10) X 7 (period t input combinations) = 560 possible shop scenarios. For each scenario, the minimum cost number of kanbans must be computed. Therefore the shop must be simulated an additional 10 times to vary the number of kanbans from 1-10. The number of kanbans that produces the minimum cost will then be included in the set of training data. The total number of simulation runs is 5600, however, only 560 data points are generated because only the optimal kanban values are included.

The sample data set generated is a representation of a multidimensional surface generated by the functional relationship between the input factors and the optimal number of kanbans. The cost of each case is based on the finished goods inventory, work-in-process inventory, and the amount of overtime needed to complete production. The BASIC program SPLIT (appendix A) randomly splits the 560 data points into two files to be used in training and testing the neural network. The files consist of 280 records each. Each record contains values for:

- Three input factors from period t-1
- One dynamic input factor from period t (changed from period t-1)
- Two input factors from period t (unchanged from period t-1)
- The number of kanbans for period t-1
- Six output factors from the end of period t-1
- The optimal number of kanbans for period t

Choosing an Appropriate Model

In the preceding section, a simulation model is used to generate dynamic shop data from an unknown and probably highly non-linear cost surface. If a simulation model is developed the obvious question is: why not use that model as a kanban predictor? Ideally, a simulation model would not be needed if actual shop data are available. The only reason a simulation model was used in this research is to control the range of variations of shop factors over a specified domain. Typically, a kanban simulator would not be available in most companies, and if available, it would be very problem-specific.

If shop data are available, a choice between a simulation model and a neural network model would have to be made. The neural network approach seems to be less cumbersome than the simulation approach in two areas. First, the neural network lacks any dependencies on shop structure. The neural network requires much less development time since no explicit knowledge of the shop is necessary. For instance, the distribution of shop factors would not have to be known. The only requirements for neural network development are a method of categorizing shop factor performance and sufficient training time. Changes in the shop (additional machines, workers, etc.) would require structural modification and adaptation for a simulation model, whereas a neural network could accommodate shop modifications through additional learning. Second, after a simulation model is constructed, a significant amount of computer time would be needed for experimentation with the model. After a neural network is developed, the recall procedure is very quick. Both the simulation and neural network approach require up-front development time, however the neural network seems to be easier to update, maintain, and actually use for estimation.

Another approach would be to use a regression model as a prediction tool. With respect to regression, Hecht-Nielsen (1990) argues that neural networks result in better functional approximations than regression techniques (given sufficient training data) in high-dimensional spaces. In order to develop a regression model, some knowledge of the nature of the underlying relationships between the input and output variables is generally needed. The dynamic interaction of shop factors may change the relationship between these variables, requiring major modifications to a regression model. Neural network development does not require the user to know as much about the input-output relationship. The network "discovers" hidden relationships during the training process. The associative ability of neural networks is another advantage of this approach. Regression cannot tolerate missing data and performs poorly with inaccurate data. A neural network is generally robust to missing or inaccurate data because it distributes the knowledge across numerous network connections.

Neural Network Development

Neural networks emulate the way portions of the brain are thought to perform. The brain is made up of many interconnected neurons composed of dendrites, axons, and a nucleus. A neuron receives stimuli from other neurons through its dendrites then sends a message across a synaptic gap to the nucleus to be processed. The chemical "strength" of the synaptic gap essentially weights the strength of each input stimulus. This gap strength or weight represents the stored memory of the brain and is determined by all past "learned" experiences. The nucleus then combines all its weighted inputs and produces an output released through its axons that is a function of the combined weighted inputs. This output then serves as an input to other neurons, even at times looping back as an input to itself.

The operation of a single neuron is modeled as a processing element (PE) in a neural network. As indicated in Figure 5, the PE may have many inputs (coming from sensory inputs, other neurons, or alternatively from the output of this neurode looping back on itself). Just as with the brain, each of these inputs is separately attenuated by a weight W_{ij}; it is this weight that represents the synaptic gap. All weighted inputs are then combined by the PE, typically by a simple summation function, i.e., $I_i = \sum_j W_{ij}X_j$. The processing element then generates an output that is a simple function of the input; a sigmoid function or a hyperbolic tangent is often used as the output transfer function. In this research, the output function (y) chosen is the sigmoid function, $y = 1/(1 + \exp(-I_i))$, where I_i is the simple summation function defined above. It should be noted that the way a processing element learns is by modification of its weights (the W_{ij}).

The development of a neural network requires the consideration of numerous criteria. The following is a list of criteria for the neural network needed for this application:

1. supervised learning
2. real numbered classifications
3. slow training time
4. fast execution time
5. complicated decision information

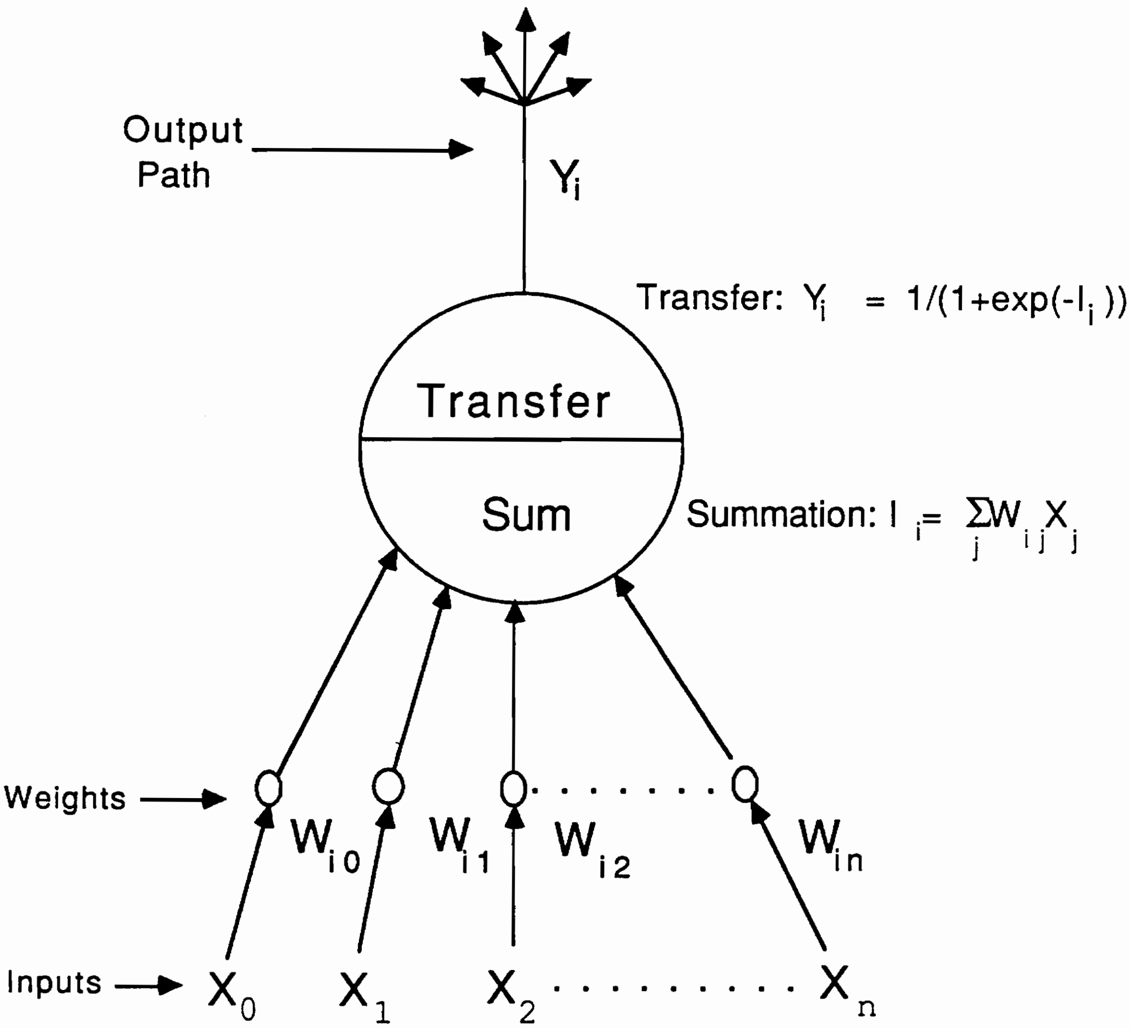


Figure 5. Processing Element

6. high information content

Based on the neural network development methodology provided by Bailey and Thompson (1990) a backpropagation paradigm was selected. Backpropagation networks consist of an input layer, an output layer, and one or more "hidden" (middle) layers of processing elements. These networks are used for ***supervised learning***, whereby training data are used to "teach" the network. In particular, each presentation of training data will include a set of inputs and the corresponding desired set of outputs. If the actual output obtained from the network by presentation of the training inputs differs from the desired output, then the weights interconnecting the neurodes are adjusted to bring the actual output closer in line to the desired output. An illustration of a backpropagation neural network like the one used in this research is provided in Figure 6.

An overview of how a "simple" backpropagation network works is now presented, complete with numerical calculations (refer to Figure 6), to give the reader unacquainted with this technique a flavor for its operation. A neural network begins processing when input data are presented to the input layer of PE's, which in turn are then sent to the hidden layer by multiplying each input by the weight W_{ij} connecting that input to the next neurode. In the particular case of Figure 6, three PE's are shown in the input layer and each is presented with an input: $X_1 = X_3 = 1$ and $X_2 = 0$. As the input layer of a backprop network serves purely as a buffer, no processing is done in the first layer, and so the output of the first layer is the same as its input, namely $X_1 = X_3 = 1$ and $X_2 = 0$. Each first-layer output is then transmitted to the two hidden-layer neurodes along the paths indicated by Figure 6, and is attenuated by the weight along the path. For example, the input to processing element 4 from the output of PE 1 is multiplied by the weight W_{41}, which is 0.5. Therefore, PE 4 re-

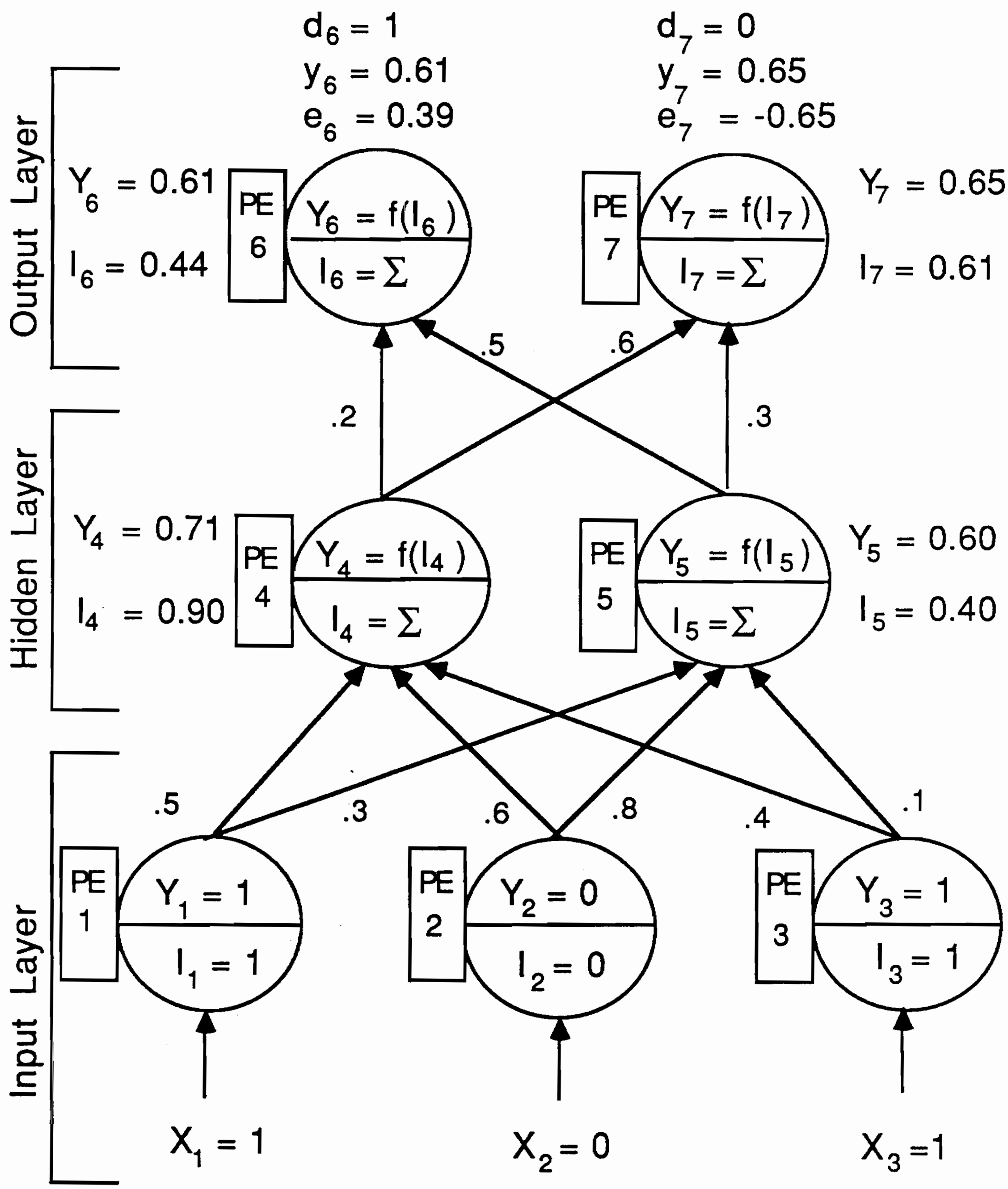


Figure 6. A Three-Layer Backpropagation Neural Network

ceives three attenuated inputs, one from each input PE, namely 0.5, 0.0, and 0.4. The weighted inputs are then summed at each neurode in the hidden layer and are subsequently modified by the neurode's transfer function. Again with respect to PE 4 in Figure 6, the PE's three inputs are summed to give 0.9. This value is fed through the sigmoid transfer function, yielding $y = 1/(1 + \exp(-0.9)) \approx 0.71$. The modified output at the hidden layer then becomes the input to the output layer. All inputs to the output layer are modified by the appropriate weights, summed, and transformed as before. The output of each output PE (0.61 at PE 6, for example) is then compared to its desired output from the training set (1.0 at that PE), and the error (i.e., desired minus actual output) is computed. In backpropagation networks, the error at each PE is ***propagated backwards*** through the network, and changes in the synaptic weights are made at each neuron to bring the network more in line with the desired output. (This portion of the network's performance is not illustrated numerically because the many scale factors introduced would tend to obscure the purpose.) The next pair of input/desired-output training data are then presented, and the process repeats, with the weights being further modified. Typically, the entire data set must be presented to the network thousands of times before the error is reduces and stabilizes. When the error is under control in such a manner, we typically say the network is "trained." Then, additional data, including data never seen by the network before, may be applied to the network and the outputs observed. Additional information on backpropagation networks is available; for example, see Rumelhart and McClelland (1986) and Klimasauskas et al. (1989).

The neural network that will be used in this research consists of an input, hidden, and output layer. The input layer consists of one processing element for each factor included in the study. A total of 13 processing elements are necessary (6 controllable variables, and 7 resultant variables).

A binary coding scheme will be used to represent the different classifications (ie. 0 = low and 1 = high) for all controllable factors. The actual level for all resultant variables will initially be included in the neural network training and recall data sets. The data will be scaled by a Min/Max table (Figure 7) in NeuralWorks. The scaling converts all input values to the range 0 to 1.

The number of PE's in the hidden layer will be 8 based on the heuristic that a good estimate of the number of PE's in the hidden layer is one-half the number of PE's in the input layer plus one ($13/2 + 1 = 7.5 \approx 8$). The number of PE's in the hidden layer will be examined further in the chapter on neural network refinements.

The output layer consists of 1 PE for the optimal number of kanbans. The optimal number of kanbans is between 1 and 10. The Min/Max table again scales the data from 0 to 1.

Refinement of the Neural Network

The neural network models used in this research must be analyzed and refined to determine the most appropriate model. Different issues on neural network development and application that must be analyzed to determine the best neural network model to use for a kanban-based production system are:

MinMax Table: NewFun # entries: 16

Col:	1	2	3	4	5
Min:	0.00	0.00	0.00	0.00	0.00
Max:	1	1	1	1	1

Col:	6	7	8	9	10
Min:	0.00	1.00	0.00	630.00	18.20
Max:	1	10	2754	5229	124.8

Col:	11	12	13	14
Min:	980.00	0.09	0.00	1.00
Max:	1159.7	0.10	80.57	8.00

Figure 7. Min/Max Table

1. What network paradigm performs best for the problem investigated?

2. What is the best hidden layer size for the network?

3. What learning/recall schedule should be used?

4. What transfer function should be used? These issues and their impacts are presented in the next chapter.

Chapter 4

Dynamically Adjusting the Number of Kanbans in a Stochastic Just-In-Time Environment

The purpose of this chapter is to analyze how a neural network can be used to adjust the number of kanbans within a JIT shop based on the dynamic interaction of critical endogenous and exogenous factors. In order to demonstrate the methodology proposed in chapter 3 data are needed on the optimal number of kanbans for shop conditions over the domain of interest. A simulation model of a JIT shop is used for generation of data under conditions of varying product demand, machine processing time, and vendor supply over a two period time frame to simulate the dynamic behavior of the shop. A post-simulation analysis optimizes over the simulation output to determine the best or "optimal" number of kanbans for each shop scenario (combination of factor levels).

The final product of the data generation phase is a data file containing all shop factors along with the optimal number of kanbans for every possible shop scenario. This data is the basis for training and evaluation of a neural network. The neural network is designed to "learn" how dynamically changing factors can be analyzed to determine the correct number of kanbans for the next production period. The network is trained on a representative random sample of half the data and then evaluated on its ability to predict the correct response for the other half of the data. A performance evaluation of the network is presented and discussed.

Once the initial network paradigm is analyzed it is refined to enhance its performance. Results of the new paradigm are compared to the original network.

Generating the Dynamic Shop Data

In order to analyze a JIT shop, a method of obtaining operating data must be determined. Due to the lack of actual shop data and the advantages of a simulation model (see Chapter 3), a SLAM II model with six workstations producing two final products is utilized. The SLAM II code is given in appendix B.

Workcenter 1 is chosen as the target workcenter for the analysis. It should be noted that any workcenter within the shop could have been chosen. Workcenter 1 was purposely made more complicated to create multiple dynamic interactions between the shop factors. The ability of the neural network to pick up these interactions and "learn" how the shop behaves is therefore a significant task.

The steps involved in generating the needed data are as follows:

1. Develop the SLAM code to represent the JIT shop
2. Develop the FORTRAN code to control the simulation process
3. Determine shop factor levels
4. Analyze using a terminating simulation
5. Determine the sample size
6. Partition the model into "runable" segments and run the simulation
7. Determine the impact of the cost function
8. Randomly split the data into two files of equal size to use for model validation

Develop the SLAM code to represent the JIT shop

The SLAM model described in the simulation section of chapter 3 was coded and debugged using SLAM II. The network consisted of 492 SLAM statements, 84 statements for workcenter 1 (the target workcenter) and roughly 55 statements for each of the other 5 workcenters. The remaining SLAM statements provide control of the simulation process.

Develop the FORTRAN code to control the simulation process

In lieu of using a separate SLAM II model for each shop scenario, a shop modification scheme allows one model to be used for all 5600 different scenarios. A control strategy written in FORTRAN modifies the shop after each run to reflect the different scenarios. The process of changing the parameters of the shop was handled by using the INIT subroutine in SLAM II. This subroutine is called at the beginning of each run and establishes the shop parameters for the controllable factors in the shop. These factors include the number of kanbans and the parameters (mean and standard deviation for demand, processing, and vendor supply) for the first period of operation. Subroutine EVENT is used to remove or add the appropriate number of kanbans between the two periods of operation and change the shop parameters for the second period of operation to create a dynamic environment. Output for the network is handled by subroutine OUTPUT. This subroutine generates a file containing the factor levels for both periods, the resultant factors from the previous period, and the total cost for the second period of operation. The cost of operation is based on the amount of overtime needed and the average level of work-in-process and finished goods inventory. A total of 492 lines of FORTRAN code are necessary as "overhead" to control parameter specification and shop operation for all possible scenarios.

Determine shop factor levels

This study investigates the impact of different levels of uncertainty for product demand, processing time, and vendor supply. In order to analyze the effect of the variation in a shop factor, the mean of the factor's distribution is held constant while the standard deviation is varied over two levels, high and low. Some initial runs were necessary to determine the range of the factor levels necessary to produce an impact on shop performance for each factor. The final levels and distribution type for each factor are:

Product demand:	Distribution	: Uniform (mean = 7)
	Low variability	: Min=7, Max=7
		: standard deviation = 0
	High variability	: Min=1, Max=13
		: standard deviation = 3.46

Processing time:	Distribution	: Log Normal (mean = 4.6)
	Low variability	: standard deviation = 0.05
	High variability	: standard deviation = 3.00

Vendor supply:	Distribution	: Discrete (mean = 5.0)

Low variability:

X	Prob(X)
4.9	.33
5.0	.33
5.1	.34

standard deviation = .00666

High variability:

X	Prob(X)
3.0	.35
3.5	.10
4.9	.05
5.1	.05
6.5	.10
7.0	.35

standard deviation = 3.251

Analyze using a terminating simulation

The purpose of the methodology is to predict the number of kanbans for the next period of operation using information from the previous period and any dynamic changes between periods. The simulation results must therefore reflect the effect of the changeover period of the dynamic factor. If the shop were allowed to run in period 2 until steady state, the changeover effect on shop performance would be nullified. Instead, the shop is run until 150 units are produced in period 2. This is a much shorter duration than is necessary for steady state. A simulation that does not reach steady state is a "terminating simulation." A terminating simulation is analogous to sampling from a probability distribution of shop performance.

Determine the sample size

The short simulation runs caused by the terminating simulations are sensitive to the random fluctuations caused by various seed values for the different random number generators. To overcome the fluctuations of different seed values for the random number streams, blocking of the experimental results is necessary to get a true representation of how a particular number of kanbans performed for a given shop scenario. The blocking was accomplished by using the FORTRAN statement LSEED to reset the random number seeds to insure the same seeds would be used across all samples.

In order to investigate the effect of the number of kanbans in a truly dynamic environment, the simulation results must reflect the result of changing only the number of kanbans for a given shop scenario. To sample from the distribution of shop performance, multiple runs for each kanban level for each scenario are needed to average out the probabilistic variations to provide a more robust measure. The question of how many samples to generate (from the distribution of shop performance) must be answered before the data can be generated. Pilot runs were performed on the network to determine the needed sample size for the data. Figure 8 shows the graphs for samples of size 5, 10, 15, and 30 for a typical data point. It can be seen that there is no significant change in the behavior of the data for any sample size greater than 5 (due to the blocking affect). Similar results were found for all other data points. Therefore, all 5600 shop scenarios were run 5 times with different seed values to provide the needed samples. Each shop factor was averaged over the five runs and the averages were written to a data file.

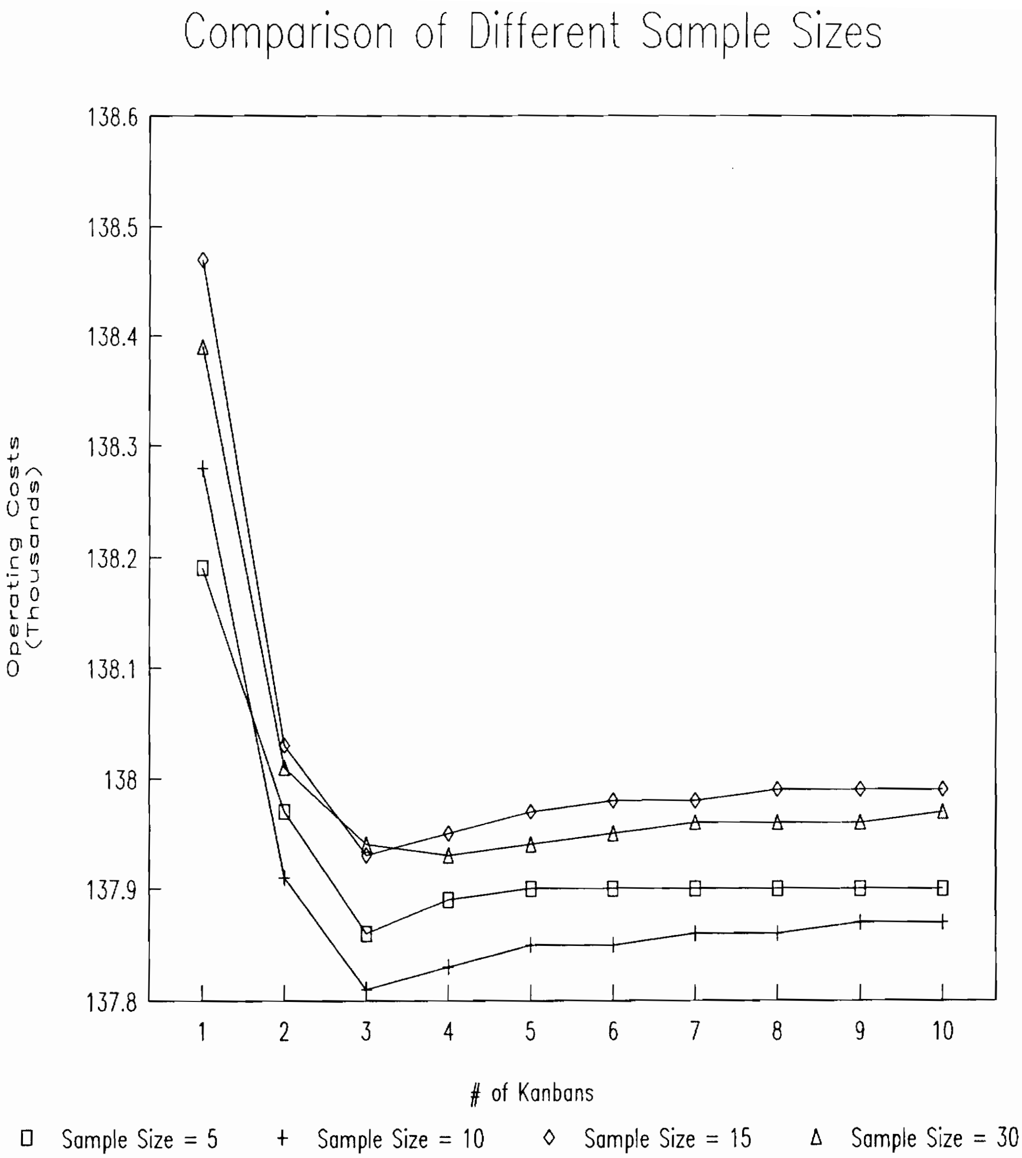


Figure 8. The affect of sample sizes of 5, 10, 15, and 30 for a typical data point.

Partition the runs into "runable" segments and run the simulation

Simulation models are well known for using massive amounts of CPU time. The initial runs indicated each data point would need around 3.45 minutes or a total of 560 X 3.45 = 1932 minutes (32.2 hours) CPU time to run the model on an IBM 3090 super computer. In order to run the model it was broken down into 7 segments of 80 data points each. The 7 simulation models were queued up and run over a period of about one week, each taking around 4.5 hours of CPU time. The 7 separate data files were transfer to an IBM PC and combined into one 800K file containing all 560 shop scenarios.

Determine the impact of the cost function

The cost function chosen is very critical since it determines the "shape" of the response surface. The cost function is based on the time to finish production of 150 units of product A and B (this represents overtime or backorder costs), and both work-in-process and finished goods inventory at the target workcenter. A basic trade-off exists between backorder and inventory costs. If backorder costs are weighted higher than inventory costs the minimum cost number of kanbans will tend to be higher to offset any shortages in inventory that could cause the production line to stop. However, the additional kanbans would add to the total cost due to the extra inventory. If inventory costs are weighted higher, the number of kanbans will tend to be lower to reduce the inventory at the workcenter, at the expense of more backorders. Given the interaction of the two factors, the cost curve should be convex. Higher total cost at both ends of the graph would result from too few kanbans causing high backorders and too many kanbans causing excessive inventory. Typically, the minimum cost will be somewhere in the middle where just enough kanbans are at the workcenter to avoid excessive backorders without a buildup of inventory.

Three cost functions are evaluated to determine the most effective method of representing the backorder versus inventory cost trade-off. The first cost function places a higher penalty on backorder costs than inventory costs. A second cost function evenly weights backorder costs and inventory costs. A third cost function places a higher penalty on inventory costs relative to backorder costs. Each cost function was used on the shop simulation data. The frequencies for the optimal number of kanbans using each function are given in Figure 9. As would be expected the first cost function resulted in a greater number of kanbans needed to prevent production flow stoppages that would result in backorders. The third cost function resulted in 1 or 2 kanbans being optimal 94% of the time. The excessive cost of additional kanbans (inventory) is allowing backorders to occur more frequently.

The third cost function is not practical for this research since the range of optimal kanbans is so tight any predictive tool should perform quite well. A close look at the effect of each cost function on a typical shop scenario is given in Figure 10. The first cost function results in a very flat curve after an initial drop in operating cost.
The cost for additional kanbans (inventory) is changing total cost by a very small amount. This flatness results in an artificially high number of kanbans for many shop scenarios since the cost for additional kanbans will sometimes drop due to the randomness of sampling. The optimal number of kanbans will typically be overestimated for the first cost function. The second cost function does a better job of picking up the cost of inventory associated with extra kanbans. This function results in a more

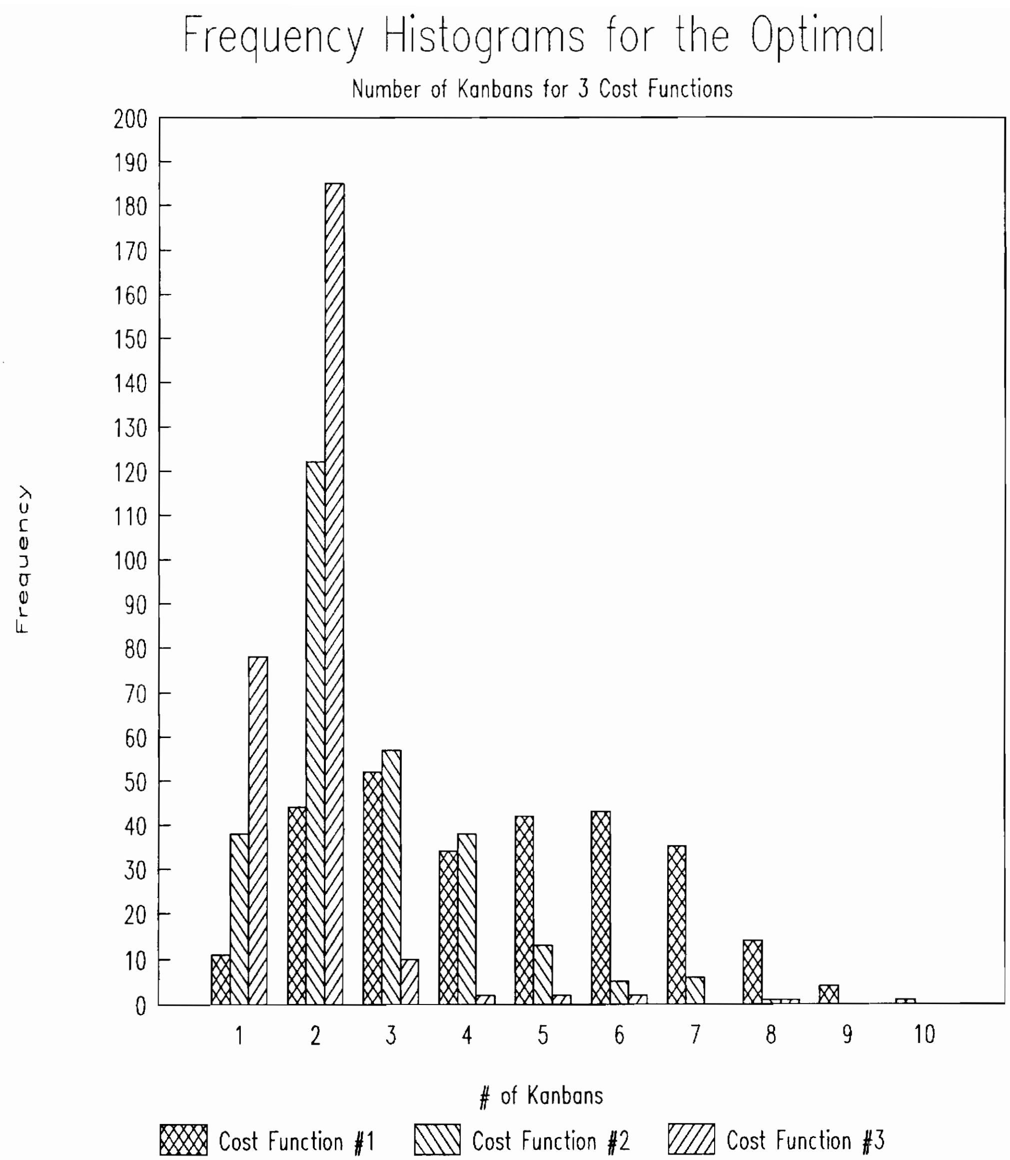


Figure 9. Frequency Histograms for the 3 Cost Functions

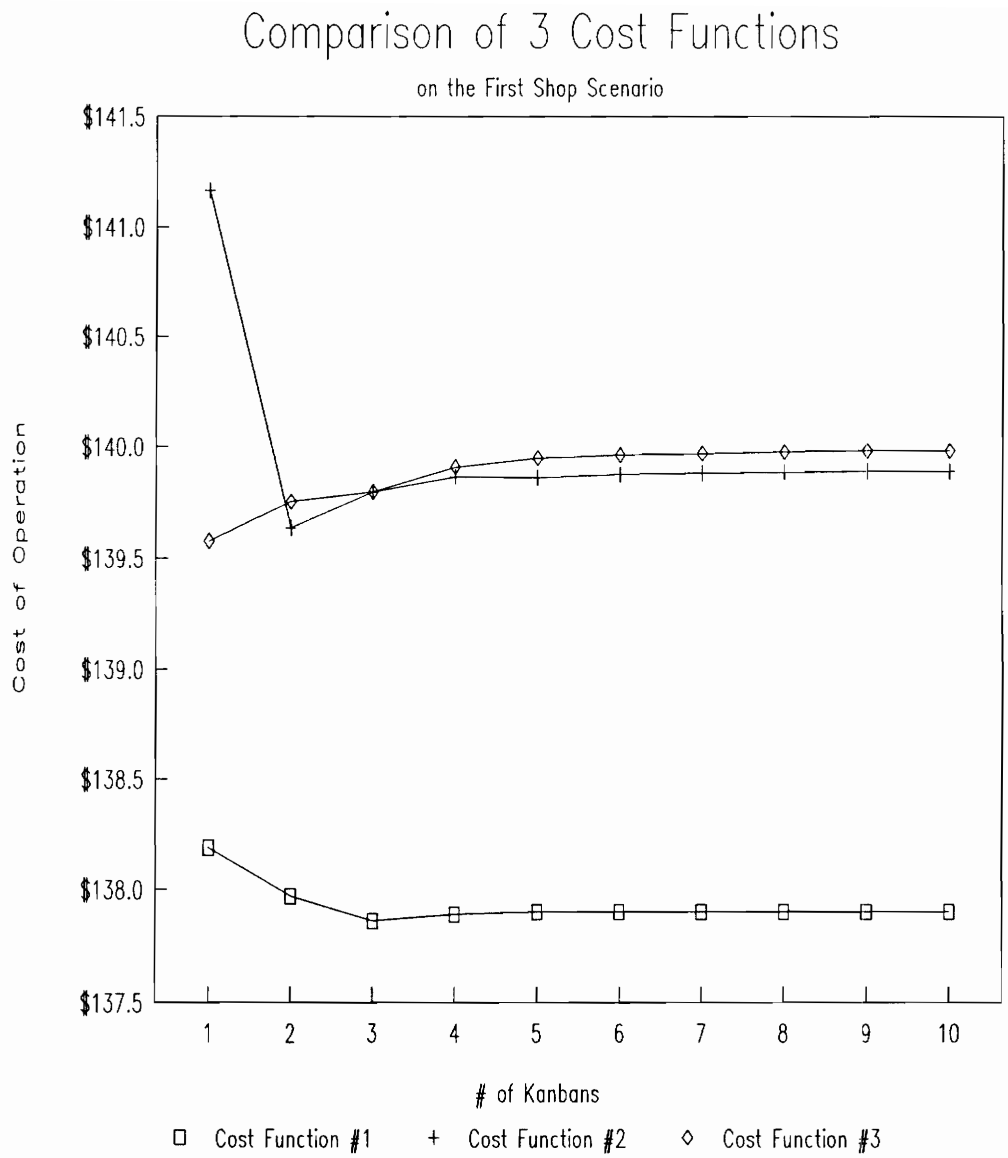


Figure 10. Graphs of 3 Different Cost Functions using the Same Data

convex curve by picking up on changes in backorder costs and inventory costs. The second cost function more clearly represents the impact of the number of kanbans at a work center and is therefore used for all subsequent analysis.

Even with the convex shape of the second cost function the predictive ability of the neural network should be carefully evaluated. If the curve is too flat, the neural network will always predict very close to the minimum cost as long as it predicts in the flat region. However, it will be difficult for the network to correctly predict the number of kanbans corresponding to the minimum cost due to the insensitivity of the cost function. The performance of the neural network in this situation must be evaluated in terms of how close the predicted number of kanbans is to the minimum cost number of kanbans. Cost should not be used in this case because it would always be artificially low. On the other hand, if the cost function produces a steep cost curve the network will be able to predict very close to the minimum number of kanbans, but any misspecification would be very costly. Therefore the neural network will be evaluated by how it predicts in terms of both cost and kanban specification. If the two evaluation schemes do not agree, the data file will be exhibiting either a very flat or steep shape. If the neural network prediction is consistent for both cost and kanban prediction, the performance of the neural network can be accurately evaluated by either method. The evaluation in subsequent sections will examine how well the network predicts in terms of cost of shop operation and the number of kanbans needed.

Randomly split the data into two equal size files to use for model validation

The simulation model was developed to produce a data file consisting of 560 data points. Each data point consists of the optimal number of kanbans for a different combination of factor levels. All 560 data points could be used as training data for the neural network, however the data was randomly split into two files of 280 records each by using the BASIC program SPLIT.BAS given in appendix A. The two files are named TR280.NNA and RE280.NNA and given in appendices C and D. The prefix "TR" designates the file used to train the network and "RE" indicates the file is to be used in the recall phase. The "NNA" extension tells NeuralWorks these are ASCII data files.

Developing the Neural Network

A standard backpropagation neural network configuration suggested by Rumelhart [86] was developed using the InstaNet submenu in NeuralWorks Professional II [89]. A printout of the network provided by NeuralWorks is given in appendix E. The neural network developed consisted of the following characteristics:

1. hetro-associative (the input data is different from the output data)
2. generalized delta rule learning
3. backpropagation control strategy
4. BFS learning/recall schedule (see Figure 11)

L/R Schedule: BFS

Recall Step	1	0	0	0
Input Clamp	0.00	0.00	0.00	0.00
Firing Density	100.00	0.00	0.00	0.00
Temperature	0.00	0.00	0.00	0.00
Gain	1.00	0.00	0.00	0.00
Modifier	2.00	0.00	0.00	0.00
Learn Step	100	200	300	400
Coefficient 1	0.50	0.10	0.05	0.01
Coefficient 2	0.20	0.05	0.01	0.00
Coefficient 3	0.00	0.00	0.00	0.00
Temperature	0.00	0.00	0.00	0.00

Figure 11. Learning and Recall Schedule BFS

5. 13 nodes in the input layer

Processing Element	Minimum	Maximum
1. Demand variability for period t-1	0	1
2. Processing variability for period t-1	0	1
3. Vendor supply variability for period t-1	0	1
4. Forecast of demand variability for period t	0	1
5. Forecast of processing variability for period t	0	1
6. Forecast of vendor supply variability for period t	0	1
7. Number of kanbans from period t-1	1	10
8. Finished goods inventory from period t-1	0	2754
9. Work-in-process inventory from period t-1	630	5229
10. Leadtime from period t-1	18	125
11. Period t-1 completion time	980	1160
12. Kanban circulation rate from period t-1 (# kanbans/completion time)	.09	.10
13. Withdrawal kanban waiting time	0	81

6. 8 nodes in 1 hidden layer

7. 1 node in the output layer

Processing Element	Minimum	Maximum
1. Number of kanbans for period t	1	8

8. linear transfer function in the input layer

9. sigmoid transfer function in the hidden and output layers

10. automatic scaling and offsetting of input and output data by a MinMax table

Training and Evaluating the Network

Before the neural network can be used to predict the number of kanbans needed for a given shop scenario it must be "trained" on past data. During the training phase the network "learns" by adjusting the weights between nodes of the network. The input data, 280 randomly selected data points in file NTR280.NNA, must be presented to the network many times. The exact number of times the data should be presented is not known. The method used to determine a stopping point for training involves using file NRE280.NNA, the other 280 data points not used in the training. The network is "saved" at many intervals and tested to see how well it can predict the optimal number of kanbans based on the weights it has learned.

The testing procedure involves two separate measures. The first measure is the average deviation from optimal cost and the second measure is the average deviation from the optimal number of kanbans. The BASIC program, EVALUATE.BAS given in appendix F, computes the performance of the network based on both measures. The "cost" measure looks at the difference between the cost of operating the shop at the number of kanbans predicted by the neural network and the cost of operating at the optimal number of kanbans for the given shop scenario. The difference in cost is calculated for every data point in the recall set and averaged over all data points to provide an overall cost measure to compare between networks. The "kanban" criterion measures the difference between the number of kanbans predicted by the neural network and the number of kanbans corresponding to the minimum cost. The deviation from the optimal number of kanbans is also averaged over all 280 data points.

The results for the original network saved at 25 different intervals are given in Figure 12. The best average deviation from optimal cost for the network is $62.60 with a standard deviation of $108.00 when the network is trained for 20,000 replications. The best average deviation from the optimal number of kanbans is .6 with a standard deviation of .955, also at 20,000 replications. The 280 data points were shown to the network around 72 times to get the best cost performance possible. The 20,000 replications required 10.25 minutes of training time on a 80386 PC with a 16 MHz processor and a 80387 math co-processor.

Training Replications	Average deviation from optimal cost		Average deviation from the optimal # of Kanbans	
	$\bar{x}$	s	$\bar{x}$	s
1000	$268.7	$324.0	1.0893	0.834
2000	121.8	182.0	0.7821	0.992
3000	73.3	123.0	0.6464	0.996
4000	77.2	128.0	0.6571	0.973
5000	77.4	129.0	0.6571	0.977
10000	75.1	123.0	0.6250	0.944
15000	72.5	121.0	0.6179	0.953
20000	**62.6**	**108.0**	**0.6000**	**0.955**
25000	65.7	114.0	0.6036	0.954
30000	65.3	114.0	0.6000	0.952
35000	67.3	116.0	0.6000	0.951
40000	67.3	116.0	0.6000	0.951
45000	67.3	116.0	0.6000	0.951
50000	68.8	117.0	0.6071	0.950
55000	68.6	117.0	0.6036	0.951
100000	70.7	124.0	0.6036	0.951
200000	74.4	128.0	0.6250	0.948
300000	75.3	130.0	0.6286	0.959
400000	76.1	130.0	0.6286	0.962
500000	75.6	130.0	0.6286	0.962
600000	78.6	135.0	0.6357	0.965
700000	76.6	131.0	0.6393	0.957
800000	73.3	130.0	0.6179	0.953
900000	73.2	126.0	0.6250	0.948
1000000	75.5	127.0	0.6321	0.947
1100000	73.2	123.0	0.6250	0.948
1200000	71.5	122.0	0.6179	0.938
1300000	73.4	123.0	0.6250	0.929
1400000	75.2	127.0	0.6250	0.933
1500000	78.7	135.0	0.6250	0.902
1600000	77.3	134.0	0.6107	0.867

Figure 12. Results From the Original Network

The total cost from period 2 operation ranges from around $138,000.00 to $157,000.00 depending on the shop conditions and the number of kanbans. The range of costs within a given shop scenario averaged $2,320.00 with a standard deviation of $2,100.00 over all 280 data points in the recall set. The maximum range was $10,046.00 and the minimum range was $430.00. Given the cost structure, an average deviation of only $62.60 from the optimal cost represents a very strong performance in terms of cost. In other words, the neural network was able to predict the number of kanbans to use in period t that on average would yield a cost only $62.60/$2,320.00= 2.7% from the optimal cost. However, as previously mentioned the network must also be evaluated in terms of how close to the optimal number of kanbans it is predicting. The average of .6 kanbans from optimal with a standard deviation of .955 shows the network is not only predicting well in terms of cost (a difficult task with a steep cost function), it is also predicting well in terms of the number of kanbans to use (difficult with a flat cost function). When additional training increases/decreases the predictive ability in terms of cost the predictive ability in terms of kanbans will generally increase/decrease. The consistent results in predictive ability indicate the cost function provides a legitimate representation of the affect of misspecifying the number of kanbans at a workcenter. It should be reiterated that in every result in Figure 12, the network was predicting in situations that it had never seen before.

Network Construction Enhancements to Improve Network Performance

In this section alternative neural network models are developed and tested to determine the most effective prediction tool. Three areas of model construction are analyzed; the network paradigm, the number of processing elements in the hidden layer, and the learning/recall schedule. The initial model described in the preceding section is used as a basis for evaluation.

First, recall the basic model is a standard backpropagation paradigm with 8 processing elements in the hidden layer and the "BFS" learning/recall schedule. The alternative network paradigm tested is the "predictive backpropagation" design suggested by Alan Lapedes [87]. This design incorporates a linear transfer function in the output layer to replace the sigmoid (squashing) function.

Second, a second rule of thumb for the number of processing elements in the hidden layer is also tested. The new rule states that roughly twice the number of elements in the input layer should be used in the hidden layer. Additional nodes are added to the hidden layer bringing the total to 31. The 23 extra nodes hold additional information about the interaction of shop factors. There are two penalties associated with adding the extra nodes. The extra nodes increase the number of calculations necessary in training the network (about twice the training time per replication), and more training replications are needed to adjust the additional weights.

Third, the standard learning/recall schedule for backpropagation networks (Klimasauskas 1989) given in Figure 13 is examined. The default backprop learning/recall schedule replaces the "BFS" learning/recall schedule used in the original network. This scheme holds the learning coefficients constant at .9 and .6 respectively. A printout of the network is given in appendix G.

The first modification tested is the learning/recall schedule. The default schedule is used in place of the "BFS" schedule. The model paradigm and the number of proc-

L/R Schedule: Backprop

Recall Step	1	0	0	0
Input Clamp	0.00	0.00	0.00	0.00
Firing Density	100.00	0.00	0.00	0.00
Temperature	0.00	0.00	0.00	0.00
Gain	1.00	0.00	0.00	0.00
Modifier	1.00	0.00	0.00	0.00
Learn Step	500	0	0	0
Coefficient 1	0.90	0.00	0.00	0.00
Coefficient 2	0.60	0.00	0.00	0.00
Coefficient 3	0.00	0.00	0.00	0.00
Temperature	0.00	0.00	0.00	0.00

Figure 13. Learning and Recall Schedule BACKPROP

essing elements in the hidden layer are unchanged from the original network. The model was trained and recalled on the same data files as the original network. The predictive ability improved to an average of $58.40 from optimal cost compared to $62.60 for the original model.

The predictive backpropagation paradigm was used to develop the next model tested. The default learning/recall schedule is used and 8 processing elements are still in the hidden layer. This network was also trained and recalled on the same data files. The performance improved to an average of $47.90 from the optimal cost. Next, the number of processing elements in the hidden layer was increased to 31. The predictive backpropagation paradigm and default learning/recall schedule are still used. This network model outperformed all other models tested. This model was able to predict the number of kanbans that resulted in an average of $38.20 from optimal cost.

The predictive backpropagation paradigm with 31 hidden layer nodes was tested with the "BFS" learning/recall schedule. The performance dropped to an average of $89.20 from optimal cost. A standard backpropagation paradigm with 31 hidden layer nodes and the default backpropagation learning/recall schedule was also tried. This network predicted to within an average of $78.40 from optimal cost.

One last network paradigm suggested by Lapedes [87] was tested to try to find the best neural network to use for predicting the number of kanbans. A cumulative backpropagation network with a linear transfer function in the output layer and weights at the hidden layer was tested. The hidden layer consists of 31 processing elements and the default learning/recall schedule was used. The network predicted within an average of $42.50 from optimal cost.

The results for the model with the best predictive performance, the predictive back-propagation network with 31 nodes in the hidden layer and the default learning/recall schedule, are given in Figure 14. This network does give a smaller average deviation from optimal cost than the original network. The smallest average deviation from optimal cost is $38.20 with a standard deviation of $91.00. The smallest deviation from the optimal number of kanbans is .4571 with a standard deviation of .810. Both are from the network trained for 1,600,000 replications. The network is able to predict with an average error of just 1.6% of the relative range of operating costs. Like the original network, the predictive ability in terms of both cost and the number of kanbans is very agreeable. The consistent results again indicate the cost function provides a legitimate representation of the affect of misspecifying the number of kanbans at a workcenter. The network required 1 hour 40 minutes of training time per 100,000 learning replications on a 80386 PC with a 16 MHz processor and a 80387 math co-processor. The network was trained an additional 400,000 replications to insure the performance had peaked. Again, in every result in Figure 12 the network was predicting in situations that it had never seen before.

Comparing the Neural Network Approach with Multiple Linear Regression

Many theoretical and empirical forecasting techniques have been devised to predict a quantitative result based on its relationship with other quantitative variables. Undoubtedly the most notable is regression analysis. The neural network approach presented in this chapter is compared to a regression approach for predicting the number of kanbans at a workcenter. In order to compare the two approaches, a re-

Training Replications	Average deviation from optimal cost		Average deviation from the optimal # of Kanban	
	$\bar{x}$	s	$\bar{x}$	s
100000	$64.9	134.0	0.5607	0.817
200000	46.5	103.0	0.4821	0.779
300000	43.4	95.0	0.4821	0.793
400000	44.6	96.0	0.4786	0.779
500000	41.4	93.0	0.4643	0.792
600000	43.9	95.0	0.4714	0.788
700000	41.9	90.0	0.4750	0.792
800000	42.1	90.0	0.4786	0.797
900000	40.4	89.0	0.4643	0.797
1000000	40.0	88.0	0.4643	0.797
1100000	38.2	85.0	0.4571	0.792
1200000	39.4	92.0	0.4607	0.792
1300000	39.8	92.0	0.4643	0.797
1400000	39.7	92.0	0.4643	0.810
1500000	38.4	91.0	0.4607	0.810
1600000	38.1	91.0	0.4571	0.810
1700000	38.5	91.0	0.4607	0.814
1800000	38.6	91.0	0.4643	0.814
1900000	38.6	91.0	0.4607	0.805
2000000	38.6	91.0	0.4607	0.805

Figure 14. Results From the Improved Network

gression equation is computed from the same date used for training the neural network. The equation parameters are then used to predict the correct number of kanbans for the shop scenarios in the recall data set used to evaluate the neural network. The performance of each approach is compared to determine which tool predicts the best in terms of operating costs.

The SAS REG procedure was used to calculate the regression coefficients for the model. The results of PROC REG are given in Figure 15. The regression equation used is:

$$
\begin{aligned}
Y = {} & -126.51 - 5.75\ ID + 0.522\ IP - 0.196\ IV + 1.23\ OD + \\
& .486\ OP + .272\ OV + .663\ KAN - .002\ INV - .0002\ WIP - \\
& .193\ LT + .129\ FIN + 35.2\ CIR + .038\ QUE
\end{aligned}
$$

ID - Demand variability for period t-1
IP - Processing variability for period t-1
IV - Vendor supply variability for period t-1
OD - Forecast of demand variability for period t
OP - Forecast of processing variability for period t
OV - Forecast of vendor supply variability for period t
KAN - Number of kanbans from period t-1
INV - Finished goods inventory from period t-1
WIP - Work-in-process inventory from period t-1
LT - Leadtime from period t-1
FIN - Period t-1 completion time
CIR - Kanban circulation rate from period t-1
QUE - Withdraw kanban waiting time

The multiple regression model's best performance is an average of $96.30 from the optimal cost compared to an average of $38.10 from optimal cost for the neural network model. In order to test the significance of the difference in prediction ability of the two models, a matched sample pairs statistical procedure is used. The test data is generated by matching the deviation from optimal cost for the neural networks model to the deviation from optimal cost for the regression model on the same shop scenario. The controlled experiment generates a data set of 280 differences (obser-

PROCEDURE REGRESSION

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	13	200.916	15.455	15.63	0.0001
Error	266	263.026	0.988		
C Total	279	463.942			

Root MSE	0.994	R-square	0.43
Dep Mean	2.514	Adj R-sq	0.41
C.V.	39.549		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob>|T|
INTERCEP	1	-126.514	42.1912	-2.999	0.0030
ID	1	-5.747	2.2731	-2.529	0.0120
IP	1	0.522	0.1771	2.947	0.0035
IV	1	-0.196	0.1510	-1.299	0.1950
OD	1	1.234	0.1203	10.258	0.0001
OP	1	0.486	0.1203	4.041	0.0001
OV	1	0.271	0.1215	2.235	0.0262
KAN	1	0.663	0.2090	3.173	0.0017
INV	1	-0.001	0.0006	-2.506	0.0128
WIP	1	-0.001	0.0002	-0.769	0.4428
LT	1	-0.192	0.0492	-3.910	0.0001
FIN	1	0.129	0.0419	3.076	0.0023
CIR	1	35.200	31.4873	1.118	0.2646
QUE	1	0.038	0.0645	0.594	0.5529

Figure 15. Results of SAS Procedure REG

vations), one observation for each shop scenario in the recall data set. The data are evaluated by the SAS procedure UNIVARIATE producing the results in figure Figure 16 The assumption of normality is automatically tested by the Shapiro-Wilk test. The hypothesis of normality is rejected (p-value $<$ .0001). There is strong evidence the data come from a non-normal distribution. However, since the number of pairs $>$ 30 ($n = 280$), the central limit theorem applies and the normality assumption is met. The test statistic for the t-test is -6.849. The probability of a greater absolute value for this statistic under the null hypothesis that the population mean is 0 (there is no significant difference between the predictive ability of the two models) is less than 0.0001. This small p-value is strong evidence that the neural network outperforms the regression model for the data tested.

REGRESSION AND NEURAL NETWORK PAIRED-COMPARISONS T TEST

Univariate Procedure

Moments

N	280	Sum Wgts	280
Mean	-0.05815	Sum	-16.2828
Std Dev	0.146359	Variance	0.021421
Skewness	-1.53309	Kurtosis	4.245735
USS	6.923325	CSS	5.976433
CV	-251.68	Std Mean	0.008747
T:Mean=0	-6.64861	Prob>|T|	0.0001
Num ^= 0	135	Num > 0	32
M(Sign)	-35.5	Prob>|M|	0.0001
Sgn Rank	-3044	Prob>|S|	0.0001
W:Normal	0.771918	Prob<W	0.0001

Figure 16. Results of SAS Procedure UNIVARIATE for a Match-pairs Test for Regression and Neural Networks

Chapter 5

A Neural Network Approach for Identifying Critical Shop Factors in a Stochastic Just-In-Time Environment

In the preceding chapter, a methodology for determining the appropriate number of kanbans to use at a particular workstation was developed for a dynamic JIT shop. This methodology uses a neural network to "learn" how to predict the number of kanbans needed for the next period of operation based on the behavior of several shop factors from the previous period and a forecast of the next period's factors. In this chapter, this methodology is extended to provide the user with insight into which dynamic factors are "critical" for smooth shop operation. The implication is that the neural network approach for predicting the number of kanbans also provides the user with the ability to identify which dynamic factor (or combination of factors) significantly impacts shop performance. In this manner, the neural network not only learns how to predict the number of kanbans for the system (operational control), but also identifies specific factors important to the efficiency of the JIT system (system design).

Utilizing the Explanatory Value of the Neural Network

In chapter 4 a neural network was developed that looked at various shop factors over a two-period horizon and predicted the number of kanbans to be used during the next production period. This same basic network can be used to identify which factors in the training data set impact on the network's ability to predict. The training data, file TR280.NNA in appendix 3, consists of 280 different observations (each observation represents a combination of shop factor levels). Each observation contains information on the level of variability of demand, processing time, and vendor supply over two periods as well as information on the number of kanbans and 6 resultant variables from the previous period of operation. The selection of which factors to include in the study was based on previous research on JIT and an intuitive feel for the chosen shop (see chapter 2). The decision was made prior to data collection and neural network training. Each factor included was "assumed" to be helpful in providing information on the necessary number of kanbans to use at a workstation. However, up to this point there was no way of knowing if the neural network's ability to predict was benefited by, unaffected by, or hindered by including a factor in the process.

To gain insight into the efficiency of the JIT system, the explanatory power of the neural network is used to identify each factor's impact on shop performance. The effect of including a factor in the methodology is felt in one of three ways. First, if the neural network's ability to predict is unchanged by removing a factor, the factor should be "trimmed" (left out of the process). Excluding a factor results in the obvious benefit of less data collection. Along with the savings in money and time in data collection, the computer processing requirements (a critical constraint when using a neural network) are reduced by deleting the nodes and connections representing the factor in the network. Second, when the neural network's prediction ability is significantly reduced by eliminating a single factor, then the factor is contributing unique and valuable information and should be kept. If a factor is critical in determining the "best" number of kanbans needed at a work center, an investigation is warranted to determine why the factor's influence is so high. Control mechanisms within the shop can be instituted for these factors. This factor analysis may prove to be the greatest benefit of the methodology since any improvement in critical resources in the production system can reap substantial benefits in overall shop efficiency. This information is very valuable to a manager implementing the JIT production system in his or her particular environment. Third, factors having a negative impact on network performance, actually hindering the network's ability to predict by presenting confounding information, should not be included. The inclusion of such a factor in neural network training is actually "confusing" the network by providing what could be described as "counter-intuitive" information. The network has trouble identifying how the information is related to the desired outputs, resulting in an adjustment to the connection weights that is detrimental to network performance. By leaving the factor out of the learning process network prediction is enhanced. Also the benefits of lower data collection costs and decreased processing requirements will again be present.

Examining the Impact of Shop Factors on Neural Network Prediction Ability

The ability of the neural network to correctly predict the number of kanbans is determined by the amount of information "learned" from the training data set. The network aggregates the information learned from the factors included in the study into weights associated with connections between processing elements. After the network is trained, it is presented new data in the form of a new combination of input factors and is asked to predict the number of kanbans. The network's prediction ability is based on the stored weights generated by the factors in the learning data set. If any one factor (or combination of factors) is eliminated from each record in the data set all information provided by the "trimmed" factor is lost. The impact on network prediction ability can be tested by comparing a network trained and recalled on trimmed data sets to a network trained and recalled on the full data set. The trimmed network's ability to predict is directly related to the amount of information lost by eliminating the factor (or factors). This procedure is directly analogous to the step-wise regression method for building a regression model.

Evaluating the Value of Dynamic Information in Kanban Prediction

The first question to address is whether the "dynamic" information (from period t-1) is providing the neural network with any benefit in terms of its prediction power. In order to evaluate the importance of the dynamic information a "static" model is created. The static model only contains information on factors in the present period. This model is totally unaware of any past shop conditions. The format for the static model follows that of the dynamic model from chapter 4. The only difference between the two is in the input layer where the static model requires only 3 nodes instead of 13. The network paradigm, number of nodes in the middle layer, learning rule, etc. are unchanged. The 3 inputs to the model are the forecast for the level of variability of demand, processing time, and vendor supply for the present period of operation. In order to create a data set for training, the training data set used for the full dynamic model was trimmed down to include only the factors from the present period (factors 4-6) and the optimal number of kanbans. This data set includes many records with the same inputs yet different desired outputs. The different outputs are the result of the impact of the previous period's ending conditions on the workcenter. Likewise, the recall data set for the static model was also trimmed down to 3 input nodes and 1 output node.

The static model was trained for the same number of replications as the dynamic model. (1,000,000). The model was saved at 10 different 100,000 intervals and recalled at each interval using the same recall file. A graphical representation of the predictive abilities of the static and dynamic models is given in Figure 17.

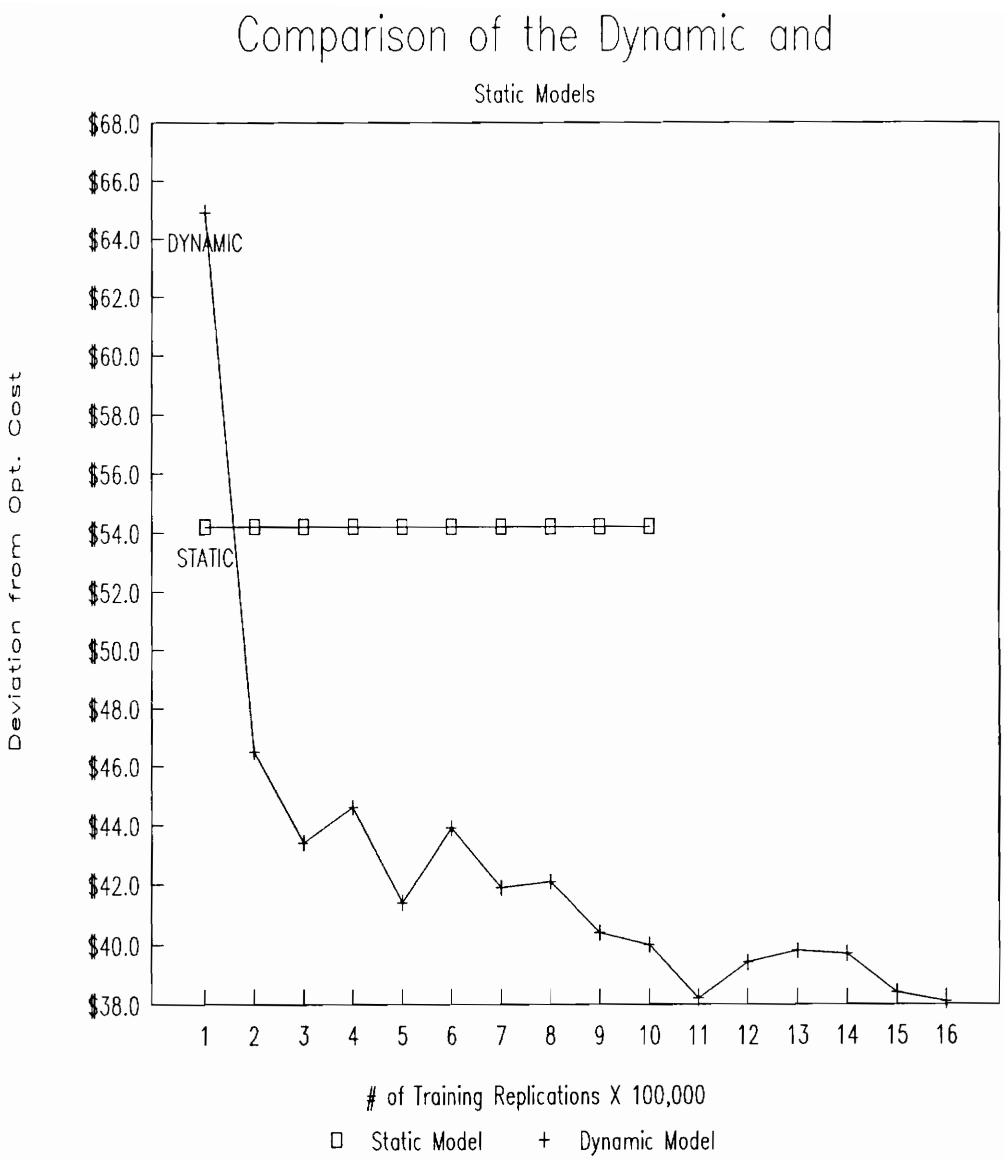


Figure 17. Comparison of the Static and Dynamic Models

The model's prediction ability quickly flattens out at 100,000 replications and never change. The static model peaked considerably sooner than the dynamic model due to only 3 nodes in the input layer. The amount of information the network could learn is limited since each factor has binary inputs representing high and low for each factor. Unlike the full network with 13 inputs (6 binary, 1 discrete, and 6 continuous) the amount of information contained in the static data set is learned with far fewer replications. The trade-off is fewer training replications (a savings in computer time) versus a diminished amount of "learned" information (poorer performance).

The static model's best performance is an average of $54.20 from the optimal cost. This appears to be a very respectable performance. The question is how does it compare to the average of $38.10 from optimal cost the dynamic model achieved? In order to test the significance of the difference in prediction ability of the two models a matched sample pairs statistical procedure is used. The test data are generated by matching the deviation from optimal cost for the dynamic model to the deviation from optimal cost for the static model on the same shop scenario. The controlled experiment generates a data set of 280 observations, one observation for each scenario in the recall data set. The data are evaluated by the SAS procedure UNIVARIATE, providing the results in Figure 18. The assumption of normality is automatically tested by the Shapiro-Wilk test. The hypothesis of normality is rejected (p-value $< .0001$). There is strong evidence the data come from a non-normal distribution. However, since the number of pairs is greater than 30 ($n = 280$) the central limit theorem applies and allows us to proceed with the test. The test statistic for the t-test is -2.3224.

The probability of a greater absolute value for this statistic under the null hypothesis that the population mean is 0 (there is no significance difference between the pre-

PAIRED-COMPARISONS T TEST

Univariate Procedure

Variable=DIFF

Moments

N	280	Sum Wgts	280
Mean	-0.01606	Sum	-4.4979
Std Dev	0.115743	Variance	0.013396
Skewness	0.391958	Kurtosis	8.350594
USS	3.809839	CSS	3.737585
CV	-720.513	Std Mean	0.006917
T:Mean=0	-2.3224	Prob>|T|	0.0209
Num ^= 0	102	Num > 0	35
M(Sign)	-16	Prob>|M|	0.0020
Sgn Rank	-888.5	Prob>|S|	0.0026
W:Normal	0.731145	Prob<W	0.0001

Figure 18. Results of the Match Pairs Test

diction ability of the two models) is 0.0209. This small p-value is strong evidence that the dynamic model does outperform the static model, hence the dynamic information is of value. This significant result shows the value of including dynamic information on changing shop conditions. The implication of this result is that dynamic information should be considered by any methodology attempting to predict the number of kanbans necessary in similar JIT systems.

Analyzing the Impact of Individual Factors on Neural Network Performance

Given the results from the previous section on the importance of the dynamic information from period t-1, an investigation on each period t-1 factor is warranted to determine its relative contribution. An investigation on each period t factor is also of interest since these factors have a direct impact on period t production. To investigate the importance of dynamic and static factors originally included in the study the following steps were taken:

1. Construct a new neural network with 1 less input processing element using the same paradigm as the full model (learning rule, transfer function, # of nodes in the hidden layer, etc.).
2. Eliminate a column of data corresponding to a single factor from the training data set and recall data set.
3. Train the new network using the trimmed file.
4. Recall using the trimmed recall data set.

5. Compute the performance of the trimmed network.
6. Repeat steps 2-5 for each of the 13 factors using the same basic model from step 1.

The procedure for generating the basic neural network used for all 13 runs was accomplished in less than 1 minute using the INSTANET function in Neural Works. The processes of trimming the data files was easily and quickly accomplished by using an editing program to delete columns of data. The name of each network consisted of "WO" for without, and a number from 1-13 corresponding to the number of the factor in the data set left out. The new networks created are WO1.NND, WO2.NND, ..., WO13.NND. The order of the 13 factors investigated is:

1. Demand Variability for Period t-1
2. Processing Variability for Period t-1
3. Vendor Supply Variability for Period t-1
4. Demand Variability for Period t
5. Processing Variability for Period t
6. Vendor Supply Variability for Period t
7. Number of Kanbans from Period t-1
8. Finished goods inventory from period t-1
9. Work-in-process inventory from period t-1
10. Leadtime from period t-1
11. Period t-1 completion time
12. Kanban circulation rate from period t-1
13. Withdraw kanban waiting time from period t-1

The network training phase required long uninterrupted learning sessions to complete the training process. Initially each network was trained up to 1,000,000 replications, pausing at intervals of 100,000 replications to save the network weights. The learning time for each network was around 1 hour 45 minutes for each 100,000 learning replications. A total of 17.5 hours were needed to train a network up to 1,000,000 replications on a 80386 PC with a 16 MHz processor and a 80387 math co-processor. A recall was performed for each of the 10 sub-networks. The 10 recalls were accomplished in about 2 minutes per network, a total of 26 minutes for all 13 networks. The decision to end the training process is based on the convergence of the graph of the each network. All networks reached their best prediction ability by 1,000,000 replications except for the networks without factors 2 and 7. Both of these networks required an additional 300,000 replications to reach their full potential. The results of training, recalling, and testing the first 4 networks (without factors 1, 2, 3, and 4) are given in Figure 19 on page 86, the next 4 networks (without factors 5, 6, 7, 8) are in Figure 20, the last 5 networks (without factors 9, 10, 11, 12, 13) are in Figure 21. In each of the previous 3 figures the full dynamic model is also plotted as a basis for comparison. Each symbol on a line in the graph represents the average deviation from optimal cost (the y value) corresponding to the number of learning replications (the x value) for a particular network. The general slope of the line connecting the points indicates the tendency of the network's ability to predict with additional training. The general trend is for a network to start high, an indication of poor performance, then converge to a certain level and flatten out. Each of the 13 networks appear to have reached their respective optimum performance.

The optimum performance of each network at the end of the training process (minimum average deviation from optimal cost) is used to compare the networks. The

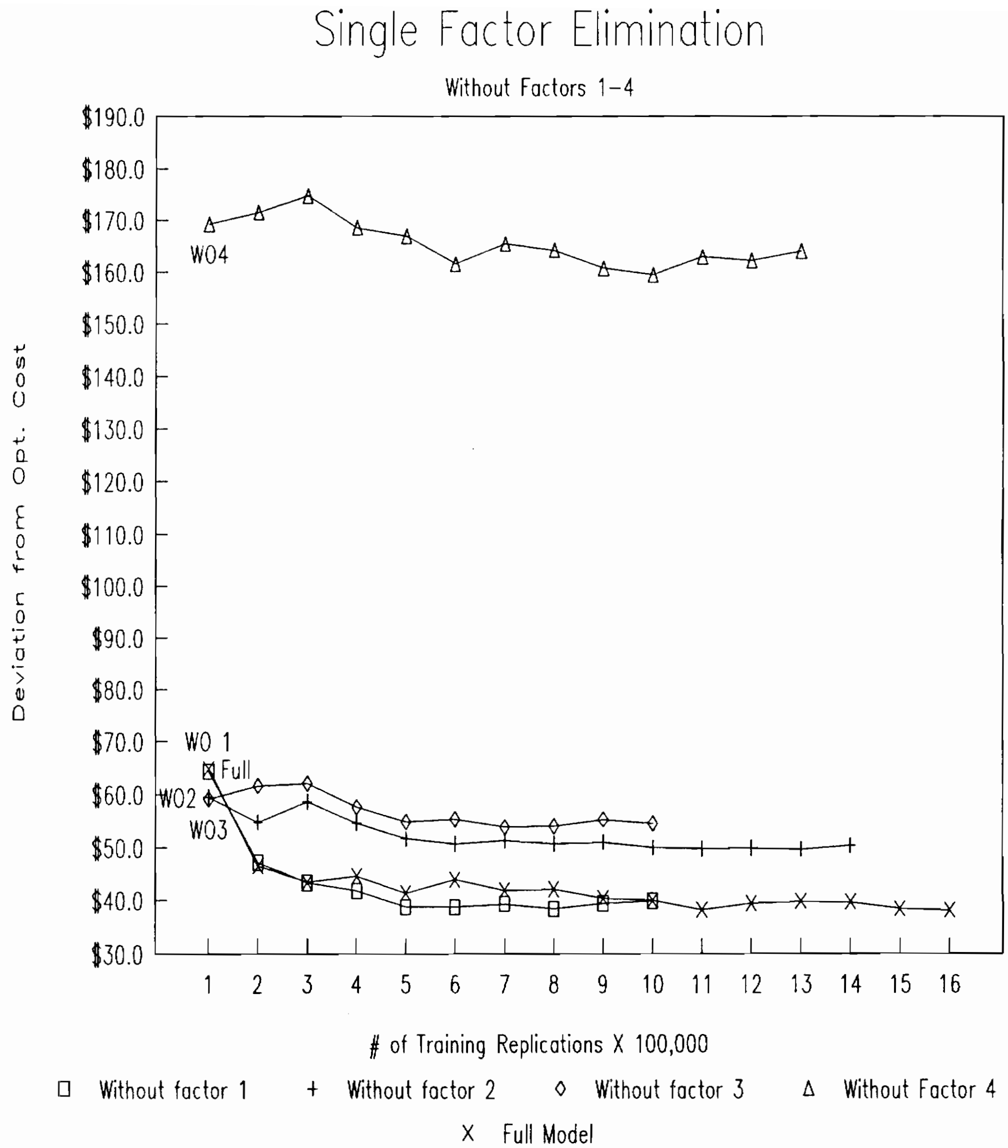


Figure 19. A comparison of Prediction Performance for Models Without Factors 1-4 and the Full Model

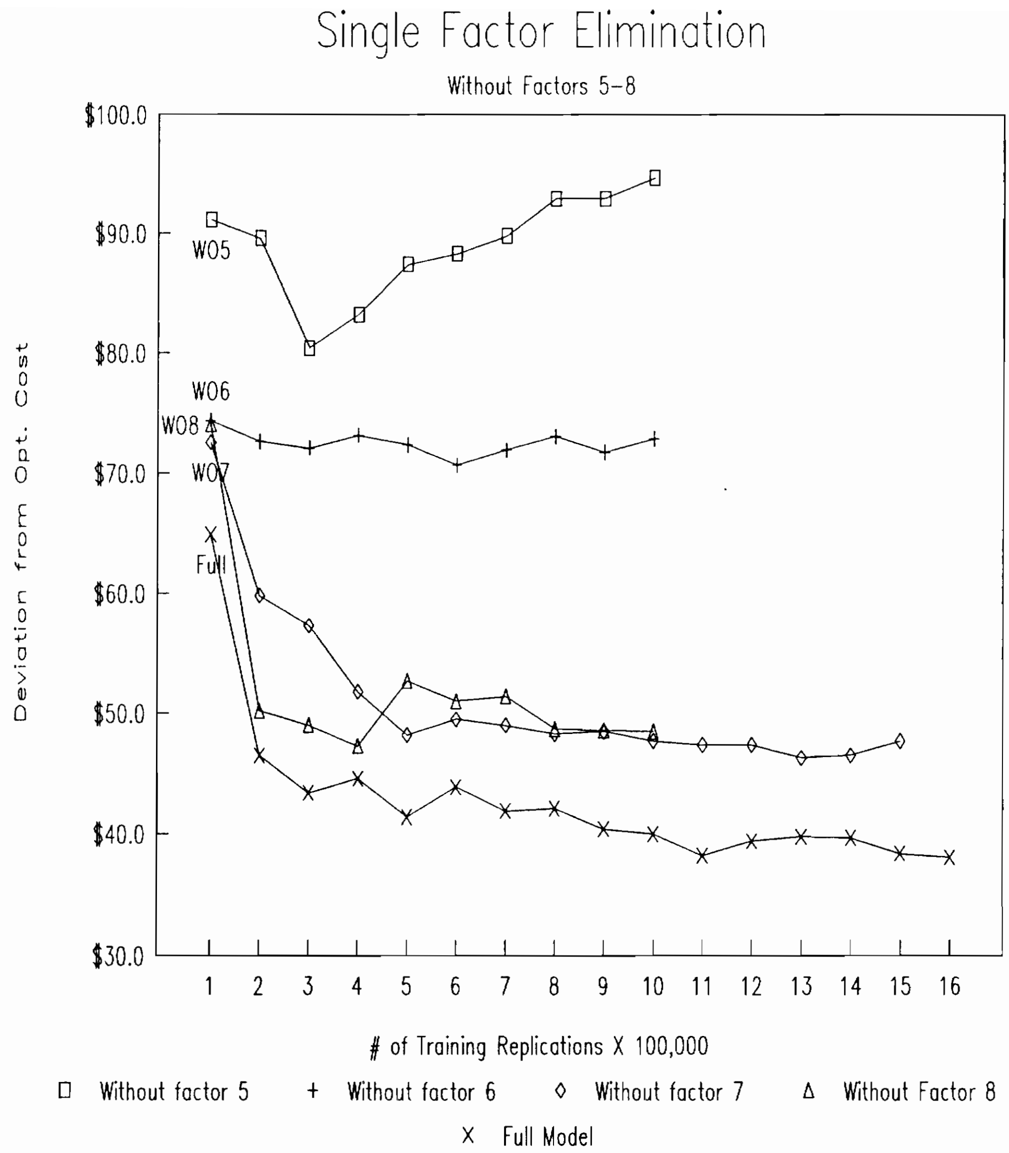


Figure 20. A comparison of Prediction Performance for Models Without Factors 5-8 and the Full Model

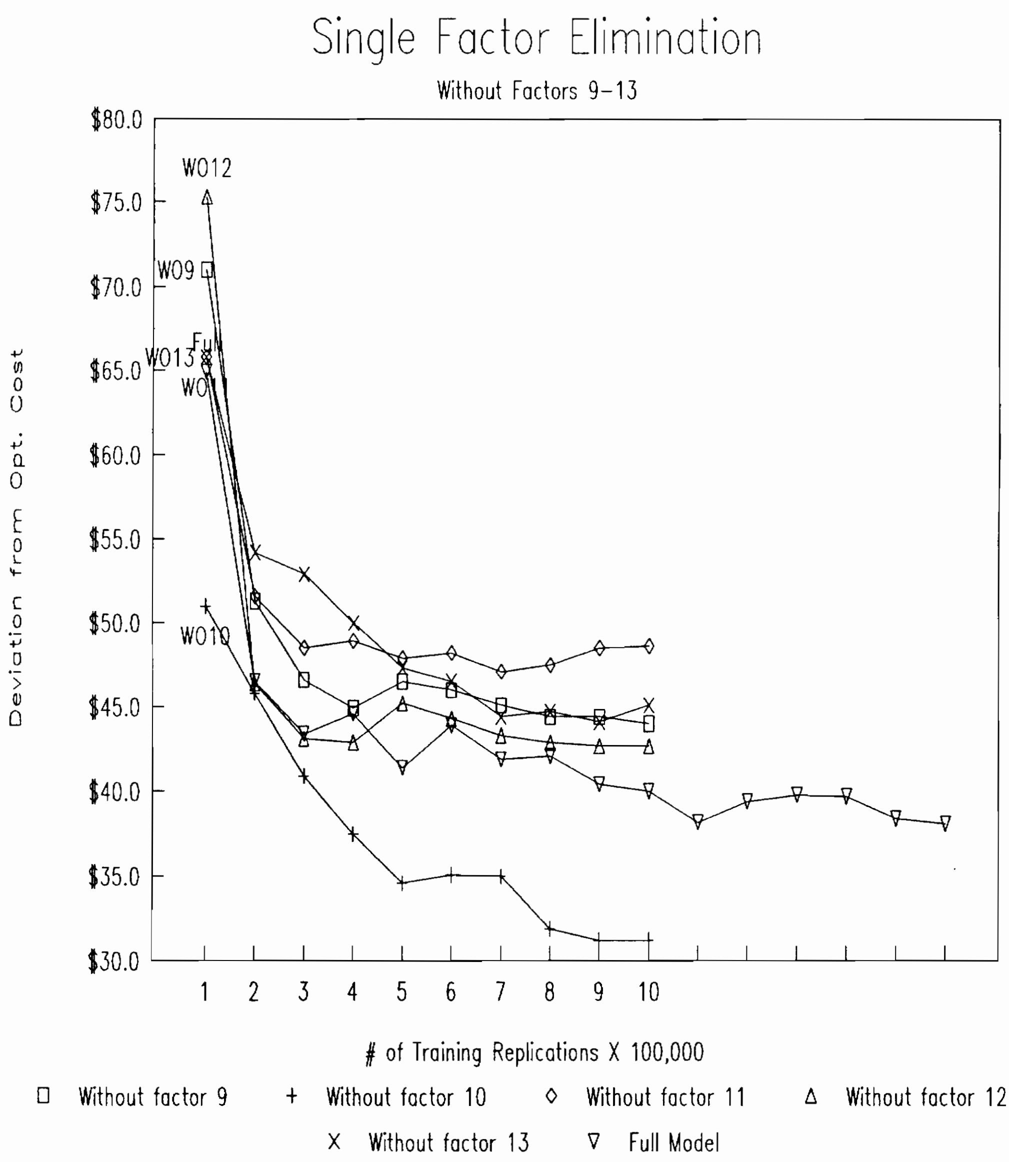


Figure 21. A comparison of Prediction Performance for Models Without Factors 9-13 and the Full Model

performance of the 13 trimmed networks as well as the full dynamic model is summarized by the bar chart in Figure 22. The models are graphed starting from left to right according to their prediction ability. The worst performer was the model without factor 4 (an average of $159.0 from optimal cost), and the best performer the model without factor 10 (an average of $31.0 from optimal cost). In order to determine whether the difference in predictive ability of the 14 networks is significant, a statistical procedure must be utilized. Like the matched pairs design of the dynamic and static models, the 14 models are recalled on the same sequence of shop scenarios. This matching constitutes a randomized block experimental design. The treatments are the elimination of individual factors while the blocks are the different shop scenarios.

Data Analysis

Since there are 14 models to test, a two-way analysis of variance (ANOVA) without interaction is the appropriate statistical model. In order to use the ANOVA model the underlying assumptions of normality and homoscedasticity must first be tested. The normality assumption is satisfied by applying the central limit theorem since all samples are of size 280. However, testing the homogeneous variance assumption presents a problem. Two common tests, the Bartlett test and Hartley test, cannot be used because of each test's sensitivity to any departure from normality within the individual samples (Neter et al., 1985). All 14 distributions exhibit very similar non-normal characteristics. A typical histogram of the deviations from optimal cost for all

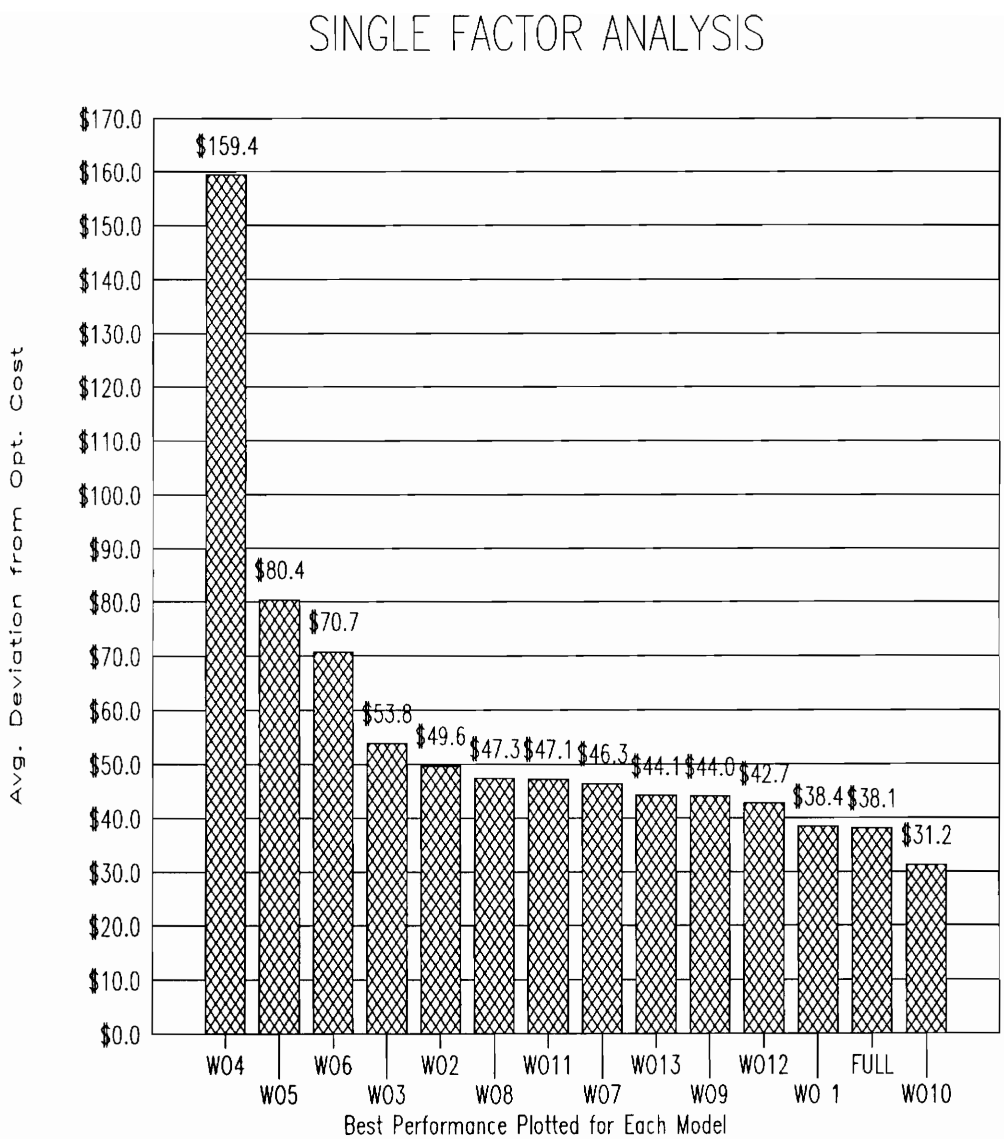


Figure 22. A comparison of Prediction Performance for All Models Trimmed by 1 Factor and the Full Model

280 shop scenarios is given in Figure 23. The results of a test for normality using PROC UNIVARIATE are given in Figure 24. The p-value $< .0001$ is strong evidence the sample comes from a population that is not normally distributed. Since the normality assumption is violated a technique that does not rely on normality must be used to test for homogeneous variance.

Layard (1973) suggests the chi-squared, box, and jackknife tests as robust alternatives for testing for homogeneous variance. The box-test is the most robust but lacks the power of the chi-squared or jackknife test. Unfortunately, none of the procedures are available in SAS and would require a great deal of programming to accomplish. Due to the difficulty in testing for homogenous variance, an alternative to the two-way ANOVA was necessary. Conover and Iman (1981) suggest a rank transformation approach for dealing with the normality problem. This procedure is a valid, powerful, and easily implemented non-parametric alternative to the parametric ANOVA approach. This approach does not require homogeneous variance and alleviates the need for a test of homoscedasticity. A non-parametric analysis is created by transforming the data into ranks and then using the ranks in a parametric ANOVA procedure. The method is implemented for the randomized complete block design by first ranking all observations from smallest to largest (regardless of block or treatment) then performing a basic two-way ANOVA on the ranked data. This procedure compares favorably with the non-parametric Friedman test and Fisher's randomization test (Iman and Conover, 1980) in terms of robustness and power.

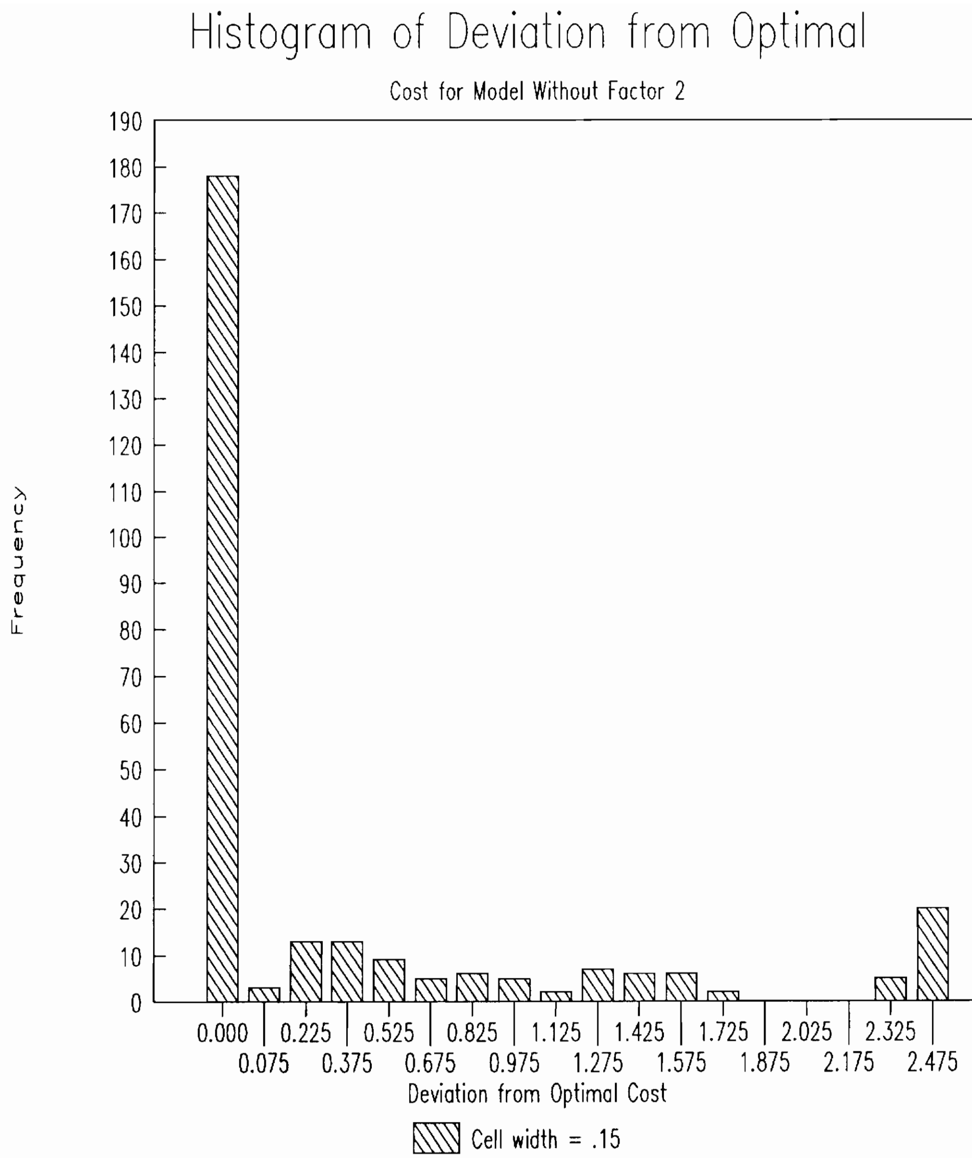


Figure 23. Histogram of of Deviation from Optimal Cost for Model WO2

TEST FOR NORMALITY FOR MODEL WITHOUT FACTOR 2

Univariate Procedure

Moments

N	280	Sum Wgts	280
Mean	0.049583	Sum	13.88328
Std Dev	0.104736	Variance	0.01097
Skewness	3.536307	Kurtosis	18.48312
USS	3.748895	CSS	3.060518
CV	211.2328	Std Mean	0.006259
T:Mean=0	7.921687	Prob>|T|	0.0001
Num ^= 0	102	Num > 0	102
M(Sign)	51	Prob>|M|	0.0001
Sgn Rank	2626.5	Prob>|S|	0.0001
W:Normal	0.562197	Prob<W	0.0001

Figure 24. Test for Normality of Deviations From Optimal Cost for Model WO2

Results of the ANOVA Procedure for Individual Factor Affects

In order to compare the prediction ability of different networks, a file is created containing one observation for each block within each treatment. The deviations from optimal cost for all 13 trimmed networks as well as the full network are stored in the SAS data file TWOWAYRANK.DAT. The data are imported into SAS and ranked by the SAS RANK procedure. The ranks are analyzed by the ANOVA procedure to test for main effects. The SAS model, TWOWAYRANK.SAS, and results of the ANOVA procedure are given in Figure 25.

The two-way analysis revealed a significant difference between the prediction ability of at least one of the 14 models (treatments) and at least one shop scenarios (blocks). The large F-value of 31.07 is strong evidence (p-value $< .0001$) of a treatment affect (the treatment being the elimination of one factor from the full dynamic model). The significant F test suggests that at least one of the factors included in the study has a significant impact on the model's prediction ability. The F-test for the blocking factor, the condition of the shop, also has a significant impact on prediction ability. The F-value of 16.42 is strong evidence (p-value $< .0001$) that blocking is important. An analysis of how different shop scenarios affect network prediction is not investigated here but is an appropriate topic for future research.

RANK TRANSFORMATION 2-WAY ANALYSIS

Analysis of Variance Procedure

Number of observations in data set = 3920

Dependent Variable: RDEV RANK FOR VARIABLE DEV

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	292	2212858143.4	7578281.3	17.08	0.0001
Error	3627	1609600893.1	443783.0		
Corr. Total	3919	3822459036.5			

R-Square	C.V.	Root MSE	RDEV Mean
0.578910	33.97962	666.17039	1960.5000

Source	DF	Anova SS	Mean Square	F Value	Pr > F
BLOCK	279	2033637065.1	7289021.7	16.42	0.0001
GROUP	13	179221078.3	13786236.8	31.07	0.0001

Figure 25. Analysis of Variance Results for the 13 Trimmed Models and the Full Model

Results of the Multi-comparison Procedure for Treatments

The previous test showed that at least one pair of models exhibited different performance. The next step is to identify where the significant differences occur. In order to compare the performance of each model to all other models, Fisher's least-significant-difference (LSD) multiple comparisons procedure is used on the rank transformed data. By using the rank transformed data the test is more robust and has more power (Conover and Iman, 1981) than a test using the raw data from non-normal populations. The experiment-wise error rate is protected at Alpha = .05 by the F test in the ANOVA. The results of the test are given in Figure 26.

An interesting observation is that the ordering of networks from worst to best performance is different when the data values are replaced with their ranks. In Figure 22 the networks are ordered according to the arithmetic mean of deviations from optimal cost. The ordering from Figure 26 is based on the arithmetic mean of ranks. There are several differences in the ordering. For instance, the order of the networks without factors 5 and 6 are reversed in the two figures. This is due to the influence of outliers in the data from the model without factor 5. These outliers are deviations of a large magnitude away from the center of the data in the positive direction. The opposite, outliers in the negative direction in the model without factor 6, is not possible since all distributions are truncated at zero. These outliers exert high influence on the arithmetic mean, inflating its value. When ranks replace the data values, the influence of these points is controlled, resulting in an average closer to the center of the data. The models without factors 2, 8, and 11 are shuffled, as well as the models without factors 9, 13, 12, and 7. However, the change in order does

```
                Fisher's Protected LSD

        Alpha= 0.05  df= 3627  MSE= 443783
            Critical Value of T= 1.96
       Least Significant Difference= 110.39


  T Grouping                 Mean       N   GROUP
            A              2621.64     280    4
            B              2143.35     280    6
            B              2096.46     280    5

            C              1960.25     280    3
     D      C              1930.08     280   11
     D      C              1924.10     280    2
     D      C              1921.77     280    8
     D      C   E          1898.86     280    9
     D      C   E          1892.31     280   13
     D      C   E          1888.73     280   12
     D      C   E          1866.98     280    7
     D          E          1822.33     280    1
                E          1805.99     280    FULL
             F              1674.16     280   10



4    6  5    3  11  2  8  9  13  12  7  1  FULL    10
A    BBBB    CCCCCCCCCCCCCCCCCCCCCCCCCC
               DDDDDDDDDDDDDDDDDDDDDDDDDD
                         EEEEEEEEEEEEEEEEEEEEE    FF
```

Figure 26. Fisher's Least Significant Difference Test

not influence any of the multiple comparisons results since all shuffled models are grouped together anyway.

Several observations can be made from the results of the protected LSD test. The first obvious result is that the model without factor 4, group "A", is significantly worse than all other networks. When factor 4 (forecast of period t demand variability) is eliminated network prediction performance is drastically reduced. This is strong evidence that demand in the present period is a very good indicator of the number of kanbans needed at a workcenter. This is not surprising since previous studies by Huang, et. al. (1983), Philipoom et. al. (1987), and others (see chapter 3) also found the amount of variability in demand to be a critical factor in the performance of JIT. However, the elimination of the demand factor from the previous period, factor 1, caused very little damage in the model's prediction ability. This seemingly contradictory information may be the result of other factors actually "covering up" for factor 1. This occurs when the information contained in factor 1 is distributed to other factors in the model. Since factors 8-13 are a direct result of the demand in the previous period (and factors 2, 3, and 7), the elimination of factor 1 is not excluding its "influence".

The prediction ability of the models without factors 5 and 6 are grouped together in group "B". These models performed better than the model without factor 4 but are significantly worse than all other models. These factors, variability of processing time and vendor supply, are also from the present period of operation. These factors were originally thought to be important since they directly impact the workcenter during the period of consideration. The "C" group consists of models without factors 3, 11, 2, 8, 9, 13, 12, and 7 while group "D" drops factor 3 and picks up factor 1. These groupings are consistent since all of these factors are from the previous period's operation. Group "E" includes the full model as well as the models without factors 9, 13, 12, 7, and 1. This is an interesting grouping since factors 9, 13, 12, 7, and 1 are in overlapping groups. Therefore, only the models without factors 3, 11, 2, and 8 from group "C" can be said to be significantly worse than the group including the full model ("E"). Each of these dynamic factors should be considered important and therefore demand close attention. Factors 3 and 2 are the variability for processing time and vendor supply for the previous period of operation. Factor 8 is the amount of finished goods inventory in the system at the end of the previous period of operation, while factor 11 is the amount of time needed to complete the previous period. The implication of this result is these dynamic factors are critical to the overall efficiency of the JIT system and require consideration from the operations manager.

The last group, group "F", consists of only the model without factor 10. This factor is leadtime from period t-1. Leadtime represents the average time an item spends at the workcenter. It is computed as the time between when an item is first demanded and when it is completed. Leadtime involves both time in queue and time in processing. However, this information is indirectly contained in other factors. Factor 13, withdrawal kanban waiting time, is one component of waiting time and should be linked to other waiting times. The mean processing time never changes but the processing time variability does (factor 2). The variation of demand (factor 1) as well as the variability of vendor supply (factor 3) also influence the congestion of the shop. The information provided by these factors eliminates the need for a measure of leadtime for the workcenter. The network prediction performance is actually better when factor 10 is not included as a training factor.

Analyzing the Impact of Factor Groupings on Neural Network Performance

The proceeding experiments have analyzed the effects of individual factors on predictive performance. Identifying groups of factors critical to efficient implementation of JIT within a shop is the goal of this section. By investigating the impact of factor groupings on the ability of a neural network to predict, critical areas within the shop can be identified. Previously, it has been shown that the group of period t-1 factors and certain individual factors are important in providing the neural network with information to assist in kanban prediction. Similarly, the impact of other groups of factors is also of concern. Six different groups of factors are chosen for study and a new network is created for each.

The first group of factors investigated are the period t-1 resultant factors, factors 8-13. These factors include finished goods inventory, work-in-process inventory, leadtime, completion time, circulation rate, and queue length. These factors are "resultant factors" since they are the result of the number of kanbans in period t-1 and the variability in demand, processing, and vendor supply in period t-1. This network is labeled "1-7 Only" and is essentially "blind" to the status of the workcenter at the end of the period t-1. The 7th factor, the number of kanbans used in period t-1, does provide some information on what the ending conditions will be. The best performance for this neural network, illustrated in Figure 27, is an average of $48.40 from optimal cost.

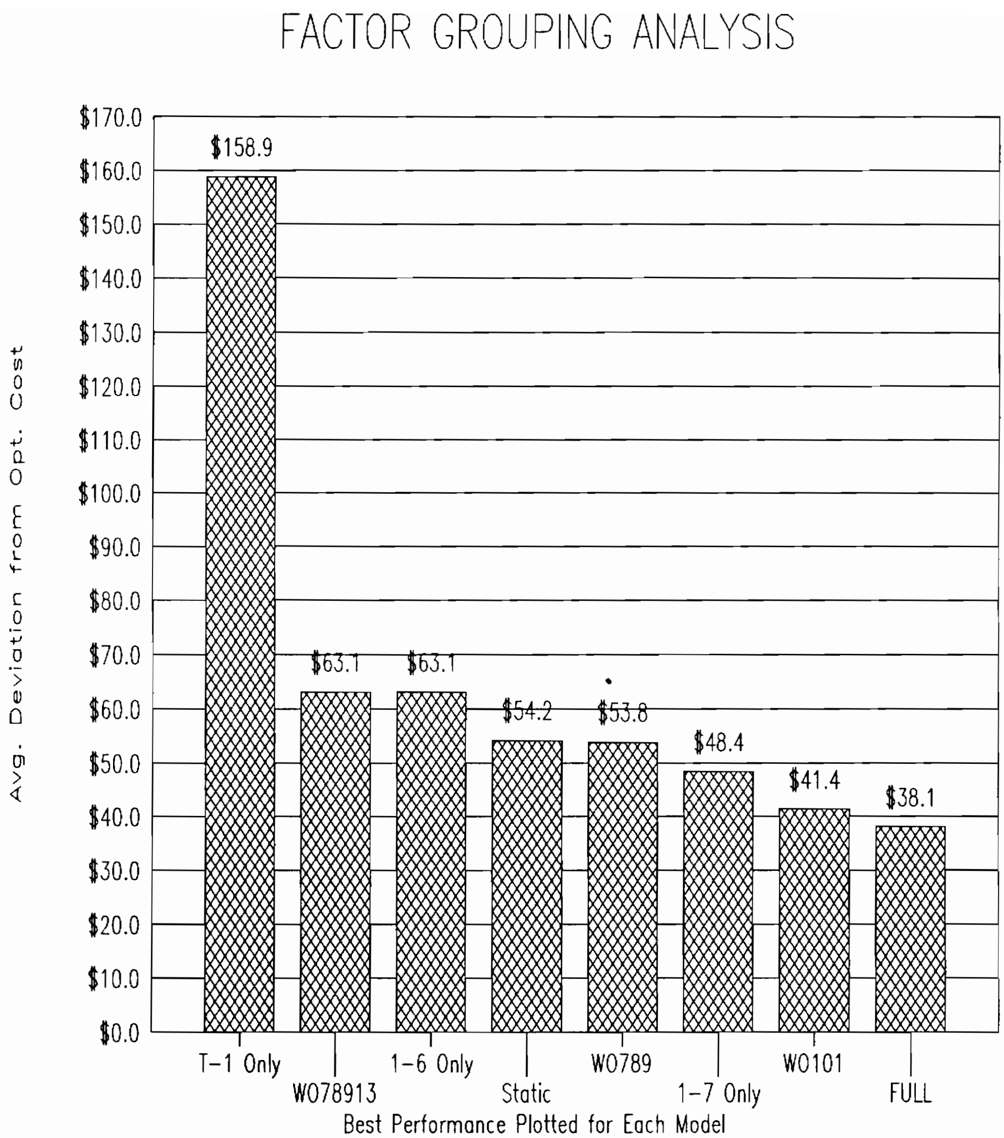


Figure 27. Factor Grouping Analysis

An interesting question is; what is the result of eliminating factor 7, the number of kanbans in period t-1, as well as factors 8-13? This network only knows the variability of demand, processing, and vendor supply for both period t-1 and the present period. This network, labeled "1-6 Only", is completely unaware of any conditions at the workcenter in terms of the inventory, kanbans, leadtimes, etc. The performance of the network in Figure 27, is an average of $63.10 from optimal cost.

A third model, "T-1 Only" in Figure 27, investigates the influence of the group of factors from the present period, the forecast for variability of demand, processing time, and vendor supply (factors 4, 5, and 6). In the preceding section each of these factors was shown to be very important in the network's ability to predict. The result of removing all three is a predictably poor average deviation from optimal cost of $158.90.

A key question for successful implementation of JIT is the effect of inventory on operating costs. The factors representing the amount of inventory at the end of the previous period of operation are grouped together in model "WO789". Factor 8 (the amount of finished goods inventory) and factor 9 (the amount of work-in-process inventory) are both directly related to factor 7 (the number of kanbans during period t-1). When all three of these factors are removed the model must predict based on limited information on ending conditions from the previous period. The performance of the network "WO789" is an average of $53.80 from optimal cost. Factor 13 was added to the group of inventory factors to form a new model "WO78913". Factor 13 represents the average waiting time of a withdrawal kanban at the workcenter and may include some of the information lost by eliminating factors 7, 8, and 9. The result on predictive ability of additionally removing factor 13 is an average of $63.10 from optimal cost.

The two factors that were individually found to be the most insignificant in terms of predictive ability were removed as a group in model "WO1&10". Recall, network predictive ability was virtually unchanged when factor 1 was eliminated and actually improved by eliminating factor 10. This model is included to test for a synergistic effect of simultaneously eliminating both factors.

Results of the ANOVA and Multi-Comparison Procedure

The differences in predictive ability of the 8 models must be tested by an analysis of variance procedure to determine if any significant differences exists. The same conditions that applied to the analysis of individual factors also apply here. Likewise, the rank transformation method is again used for the 2-way ANOVA model and multi-comparison procedure. The results of the two-way analysis are given in Figure 28. The analysis revealed a significant difference between at least one of the 8 models (treatments) and at least one shop scenario (blocks).

Fisher's protected LSD test is used to make multi-comparisons among the 8 network models. The "A" grouping consists of the model "T-1 Only". This model's predictive ability is significantly worse than all other models tested. This is a consistent result with the findings of the preceding section on individual factors. Factors 4, 5, and 6 from the present period were all found to be important in network prediction. Consequently, eliminating all of these factors should result in a model that exhibits poor performance. This result serves to confirm the belief that the forecasts for variability of demand, machine processing, and vendor supply for the present period are essential for efficient kanban prediction. The results of the test are in Figure 29.

Ranked Factor Groups

Analysis of Variance Procedure

Number of observations in data set = 2240

Dependent Variable: RDEV   RANK FOR VARIABLE AVEDEV

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	286	424679332.69	1484892.77	9.18	0.0001
Error	1953	315884213.31	161743.07		
Total	2239	740563546.00			

R-Square	C.V.	Root MSE	RDEV Mean
0.573454	35.89227	402.17293	1120.5000

Source	DF	Anova SS	Mean Square	F Value	Pr > F
BLOCK	279	381072166.81	1365850.06	8.44	0.0001
GROUP	7	43607165.88	6229595.13	38.52	0.0001

Figure 28. Analysis of Variance Results for Factor Groupings

Ranked Factor Groups

Analysis of Variance Procedure

T tests (LSD) for variable: RDEV

```
Alpha= 0.05  df= 1953  MSE= 161743.1
    Critical Value of T= 1.96
 Least Significant Difference= 66.66


T Grouping              Mean      N  GROUP

          A          1472.49    280   T-1 Only


          B          1127.11    280   1-6 Only
          B
  C       B          1109.08    280   Wo 78913
  C       B
  C       B          1105.22    280   Static
  C       B
  C       B   D      1078.81    280   Wo 789
  C           D
  C       E   D      1051.66    280   1-7 Only
          E   D
          E   D      1024.95    280   Wo 10&1
          E
          E           994.69    280   Full
```

```
T-1  1-6  Wo78913  Static  Wo789  1-7  Wo10&1  Full
AAA  BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCC
                           DDDDDDDDDDDDDDDDDDDDDD
                                    EEEEEEEEEEEEEEEEEEEE
```

Figure 29. Fisher's Least Significant Difference for Factor Groupings

The "B" grouping consists of the models "1-6 Only", "WO78913", "Static", and "WO789". This group of models was found to be significantly worse than the full model. Previously the static model was tested against the dynamic model with a consistent result. The models grouped with the static model also contain very little dynamic information. Each model includes the variability of demand, processing time, and vendor supply from the previous period (factors 1-3), but varies in terms of the resultant information included. No information on the number of kanbans, ending inventory, or work-in-process is contained in any model. However, model "WO789" contains information on leadtime, finishing time, circulation time, and withdraw kanban waiting time from the previous period. This implies the important "dynamic" information is contained in the factors relating to the inventory in the shop, factors 7, 8, and 9. Group "C" shows that the model "1-7 Only" with the number of kanbans from the previous period (factor 7) is included with the models from group "B" and model "1-6 Only" drops out. The inclusion of factor 7 does improve predictive ability over "1-6 Only" by adding some estimate of the inventory at the end of the period. However, this additional information on the inventory at the end of the previous period is not substantial enough to separate "1-7 Only" from the other models in group "C". One interesting observation from group "C" is that the "Static" (factors 4-6) is included while "1-6 Only" did not perform well enough to be included. However, since both models are included in group "B" any difference in predictive ability is presumably due to random sampling error.

The "E" grouping consists of the full dynamic model, "Full", and models "WO1&10" and "1-7 Only". The first interesting point here is that the model without factors 1 and 10 did not out-perform the full model; hence there is no synergistic effect by eliminating both factors simultaneously. The other interesting point is the model with factors 1-7 was found not significantly different from the full model. The dynamic information lost by eliminating all resultant information (factors 8-13) is offset by including the number of kanbans from the previous period. This implies a sufficient amount of dynamic information is contained in factors 1, 2, 3, and 7. This is an intuitive result since the "resultant" factors are a consequent of these inputs. The least significant difference for the test is 66.66 using an alpha level of 0.05. The difference between the "Full" model and the "1-7 Only" model is 56.97. The judicious application of this finding is left to the operations manager.

Chapter 6

Conclusions

Two problems were addressed in this research. The first problem analyzed is: how can the number of kanbans at a workcenter be predicted in a stochastic JIT shop? A methodology was developed using a neural network to learn how endogenous and exogenous factors are interrelated to the optimal number of kanbans needed at a workcenter. A neural network model was constructed and trained on data from a simulation model of a stochastic JIT shop. The network was able to learn the complex interrelations between 13 input factors and shop performance. The network was able to predict with an average error of just 2.7% of the relative range of operating costs.

Three areas of network design were identified to improve the predictive performance of the network. First, a linear transfer function was used in the output layer to replace the sigmoid function in the standard backpropagation model. The linear transfer of information between nodes eliminates the "squashing" feature inherent with the sigmoid. Second, the number of nodes in the hidden layer was increased to 31 to allow for more information on the interaction of factors to be stored. Third, the default learning/recall schedule with constant learning coefficients replaced the decreasing coefficients of the previous schedule. The new model required more processing time but consistently out-performed the original model. The average error of prediction dropped to 1.6% of of the relative range of costs. The neural network was compared to multiple linear regression to determine which tool performed the best for the dynamic shop data. A matched-pairs t-test found the neural network significantly out-performed the regression model (p-value $< .0001$). The notable predictive performance of the network is evidence that a neural network can be used in a dynamic production environment to adjust the number of kanbans, allowing efficient implementation of a JIT system.

The second problem is: how can individual factors or groups of factors critical for successfully implementing JIT in a stochastic environment be identified? The methodology for predicting the number of kanbans was extended to attack the problem. A matched-pairs test was used to determine if a neural network with dynamic information can predict significantly better than a neural network without any knowledge of dynamic factors. The dynamic model was found to significantly out-predict the static model (p-value $< .0001$). The dynamic factors provide additional information that is beneficial to the network.

An investigation of the impact of each factor (factor screening) was accomplished by eliminating a factor from the learning process and then comparing the predictive ability of the trimmed network to the full dynamic model. A two-way analysis of variance without interaction statistical model was used to test for differences in predictive ability of the trimmed neural networks.

The ANOVA procedure found a significant difference between the predictive ability of at least one of the models (p-value < .0001). Fisher's least-significant-difference multiple comparisons procedure was used to identify where the significant differences occurred. Several observations can be made from the results of the protected (by the F-test) LSD test. The variability of demand in the present period was found to have the greatest impact on predictive performance. However, the impact of variability of demand from the previous period was not found to be significant. This seemingly contradictory information may occur because the information contained in this factor is being spread over other factors in the shop. The other two factors from the present period, processing variability and vendor supply variability, also significantly impact the predictive ability of the network. Therefore, the forecasts for the three factors studied in the present period are very important in determining the correct number of kanbans at a workcenter and should be given considerable attention.

The dynamic factors from the previous period found to significantly impact network predictive performance are processing variability, vendor variability, finished goods inventory, and time to complete production (overtime). Each of these factors should be considered important in kanban specification. The implication of this result is that an operations manager should consider not only the present operating conditions, but also take into account the dynamic information from the previous period of operation.

The leadtime factor from the previous period was found to be "significantly unimportant" in network predictive ability. The network was able to predict better when this factor was completely left out. The factor is actually providing confusing information to the network and consequently hurting performance.

Seven factor groupings were investigated in an attempt to find critical areas that are important in shop operation. The test groups (with the model name in quotes) are:

1. All forecasted factors in period T (4-6), "T-1 Only"
2. The number of kanbans in period T-1 (factor 7) and the resultant factors from period T-1 (8-13), "1-6 Only"
3. The number of kanbans in period T-1 (factor 7), resultant factors dealing with inventory levels (8 and 9), and queue length (factor 13), "Wo 78913"
4. All Dynamic information from period T-1 (factors 1-3, 7-13), "Static"
5. The number of kanbans from period T-1 (factor 7) and resultant factors dealing with inventory levels (8 and 9), "Wo 789"
6. All resultant factors (8-13) from period T-1, "1-7 Only"
7. Demand variability from period T-1 (factor 1) and leadtime (factor 10) from period T-1, "Wo 1&10"

A 2-way ANOVA model found that the predictive performance of at least one of the neural network models is significantly different from the other models (p-value < .0001). Fisher's LSD is again used to make multi-comparisons among the eight networks.

Model "T-1 Only" was found to predict significantly worse than all other models. This is a consistent result with the findings from the proceeding section on individual factors. The factors from the present period, 4-6, are very important in determining how many kanbans are needed at a workcenter.

Models "1-6 Only", "Wo 78913", "Static", and "Wo 789" were found to predict significantly worse than the full dynamic model. These models differ in the amount of resultant information contained. However, none of the models contain any information on the number of kanbans from the previous period, finished goods inventory, or work-in-process inventory. This implies the important dynamic information is contained in the factors relating to the inventory in the shop. The importance of inventory is consistent with the philosophy of JIT.

The predictive ability of the full model was not significantly different from that of models "Wo 1&10" and "1-7 Only". The elimination of the two most insignificant individual factors did not produce any synergistic affect. The inclusion of factor 7, the number of kanbans from the previous period, to model "1-6 Only" did produce an interesting result. This implies a sufficient amount of dynamic inventory information is contained in factor 7. This is an intuitive result because the resultant inventory factors are a consequence of the number of kanbans at the workcenter.

Areas for Future Research

Several areas of future research can be identified. The following list suggests various extensions of this work:

1. Extend the analysis back i periods, where $i > 1$.
2. Identify correlated factors. Examine the effect of leaving out a factor that is "contained" in other factors.

3. Determine how all information from the simulation model can be used. Presently only the optimal number of kanbans for each shop scenario is used. All other information on non-optimal numbers of kanbans is eliminated.
4. Expand the scope to include more production environments such as group technology, etc.

Bibliography

Bailey, D., and D. Thompson, "How to Develop Neural Networks," *AI Expert*, (June 1990), 38-47.

Bitran, G., and L. Chang, "A Mathematical Programming Approach to a Deterministic Kanban System," *Management Science*, 33, 4 (1987), 427-442.

Burr, D., "Experiments With a Connectionist Text Reader," *Proceedings of the First International Conference on Neural Networks*, M. Caudill and C. Butler, San Diego, CA, 4 (1987), 717-724.

Chaudhury, A. and A. Whinston, "Towards an Adaptive Kanban System," *International Journal of Production Research*, 28, 3 (1990), 437-458.

Deleersnyder, J., T. Hodgson, H. Muller, and P. O'Grady, "Kanban Controlled Pull Systems: An Analytic Approach," *Management Science*, 35, 9 (1989), 1079-1091.

Dietforide, T., and R. Michalski, "Discovering Patterns in Sequences of Events," *Artificial Intelligence*, 25, 2 (1985), 187-232.

Esparrago, R., "Kanban," *Production and Inventory Management*, 29, 1 (1988), 6-10.

Forsyth, R., and R. Rada, *Machine Learning: Applications in Expert Systems and Information Retrieval*, Halsted Press, New York, 1986.

Groenevelt, H., and U. Karmarkar, "A Dynamic Kanban System Case Study," *Production and Inventory Management*, 29, 2 (1988), 46-50.

Gupta, Y. and M. Gupta, "A System Dynamics Model for a Multi-stage Multi-line Dual-card JIT-kanban System," *International Journal of Production Research*, 27, 2 (1989), 309-352.

Hecht-Nielsen, R.,*Neurocomputing*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

Hopfield, J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences of the United States of America* , 79 (1982), 2554-2558.

Huang, P., L. Rees, and B. Taylor, "A Simulation Analysis of the Japanese Just-in-Time Technique (with Kanbans) for a Multiline, Multistage Production System," *Decision Sciences*, 14, 3 (1983), 326-344.

Kinoshita, J., "Neural Networks at Work," *Scientific American*, (November 1988), 134-135.

Klimasauskas, C., J. Guiver, and G. Pelton, *NeuralWorks Professional II and NeuralWorks Explorer, Volume 1, Neural Computing,* NeuralWare, Inc., Pittsburgh, PA, 1989.

Korf, R., "Macro-Operators: A Weak Method for Learning," *Artificial Intelligence*, 26, 1 (1985) 35-77.

Krajewski, L., B. King, L. Ritzman, and D. Wong, "Kanban, MRP and Shaping the Manufacturing Environment," *Management Science*, 33, 1 (1987), 39-57.

Lapedes, A., and Farber, R., "Non-Linear Signal Processing using Neural Networks: Prediction and System Modeling" Los Alamos National Laboratory report LA-UR-87-2662, 1987.

Lee, S., and M. Ebrahimpour, "Just-In-Time Production System: Some Requirements for Implementation," *International Journal of Operations and Production Management*, 4 (1984), 3-15.

Minsky, M., and S. Papert, *Perceptrons*, MIT Press, Cambridge, Massachusetts, 1969.

Monden, Y., *Toyota Production System*, Industrial Engineering and Management Press, Norcross, Georgia, 1983.

Neter, J., Wasserman, W., and M. Kutner, *Applied Linear Statistical Models*, Irwin Press, Homewood, Illinois, 1985.

Philipoom, P., L. Rees, and L. Wiegman, "Using Neural Networks to Determine Internally-Set Due-Date Assignment "Rules" for Job-Shop Scheduling," working paper, R. B. Pamplin College of Business, Virginia Tech, (1990).

Philipoom, P., L. Rees, and B. Taylor, "Simultaneously Determining the Number of Kanbans, Container Sizes, and the Sequence of Products in a Just-in-Time Shop," under review, (1990).

Philipoom, P., L. Rees, B. Taylor, and P. Huang, "An Investigation of the Factors Influencing the Number of Kanbans Required in the Implementation of the JIT Technique with Kanbans," *Management Science*, 25, 3 (1987) 457-472.

Pritsker, A., C. Sigal, and R. Hammesfahr, *SLAM II: Network Models for Decision Support*, Prentice-Hall, Inc., New Jersey, 1989.

Ragatz, G., and V. Mabert, "A Simulation Analysis of Due Date Assignment Rules," *Journal of Operations Management*, 5, 1 (1984), 27-39.

Rakes, T., L. Rees, F. Siochi, and B. Wray, "Estimating the Number of Kanbans Using Neural Networks", *Advances in AI in Finance, Marketing, and Management*, (1992).

Rees, L., P. Philipoom, B. Taylor, and P. Huang, "Dynamically Adjusting the Number of Kanbans in a Just-In-Time Production System Using Estimated Values of Leadtime," *IIE Transactions*, 19, 2 (1987), 199-207.

Rees, L., P. Philipoom, and T. Rakes, "Dynamically Setting the Number of Kanbans in a Just-In-Time Shop: an AI Approach Using Machine Learning," R.B. Pamplin College of Business Working Paper No. VT AIMS-LPR-87-4, Virginia Tech, (1987).

Rosenblatt, F., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychoanalytic Review*, 65 (1958), 386-408.

Rumelhart, D., and J. McClelland, *Parallel Distributed Processing*, MIT Press, Cambridge, Massachusetts, 1986.

Sejnowski, T., and C. Rosenberg, "Parallel Networks that Learn to Pronounce English Text," *Complex Systems*, 3 (1987), 145-168.

Schonberger, R., "Just-In-Time Production Systems: Replacing Complexity With Simplicity In Manufacturing Management," *Industrial Engineering*, (1984).

Waters, C., "Why Everybody's Talking About "Just-In-Time"," *INC*, (March 1984).

Appendix A

BASIC Program SPLIT.BAS

```
 5 REM  SPLIT.BAS - this program randomly splits the data
 8 REM  into 2 files of equal size. One for training and one
 9 REM  for recall.
10 RANDOMIZE 4589
20 CLS
30 T1=0
40 T2=0
50 T3=0
60 T4=0
70 T5=0
80 T6=0
90 T7=0
100 T8=0
110 T9=0
120 T10=0
130 OPEN "TR280.NNA" FOR OUTPUT AS #1
140 OPEN "RE280.NNA" FOR OUTPUT AS #2
150 OPEN "CALL.NNA" FOR INPUT AS #3
160 CONUT1=0
170 CONUT2=0
180 FOR K = 1 TO 560
190 INPUT #3,L$
200 RNUM=RND
210 IF RNUM> =.5 THEN PRINT #2,L$:COUNT2=COUNT2+1
220 IF RNUM<.5 THEN PRINT #1,L$:COUNT1=COUNT1+1
230 NEXT K
240 PRINT COUNT1,COUNT2
250 STOP
```

Appendix B
SLAM II Model of a JIT Shop With 6 Workcenters

```
//CC00 JOB 54548,'B A WRAY',TIME=5,REGION=2048K
/*PRIORITY IDLE
/*ROUTE PRINT VTVM1.BAW
/*JOBPARM LINES=30
//STEP1 EXEC SLAMCG,PARM.FORT='OPT(3)'
//FORT.SYSIN DD *
C SHOP SIMULATOR
C*************************************************************
      DIMENSION NSET(50000)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/GCOM5/ IISED(10),JJBEG,JJCLR,MMNIT,MMON,NNCFI,
     1NNDAY,NNPT,NNRNS,NNSTR,NNYR,SSEED(10),LSEED(10)
      COMMON QSET(50000)
      COMMON/KANS/IOLD,INEW,NUMADD,NUMDEL
      COMMON/SCOM9/CPROB(9),VALUE(9)
      COMMON/FACTORS/IF1,IF2,IF3,JF1,JF2,JF3,ISAMPLE
      COMMON/SAVES/SCOST,IK2,SAVG,SINV
      COMMON/SAVE2/SOINV,SOWIP,SOLEADT,SORES,SOCIR,SOWITHD,SOPROD,SOFIN
      COMMON/PPS/PPINV(30),PPWIP(30),PPLEAD(30),PPFIN(30),PPCIR(30)
      COMMON/PP2/PPRES(30),PPWITHD(30),PPPROD(30)
      COMMON/COSTS/AVGLEADA(30),AVGLEADB(30),TOTINV(30),WIPS(30)
      COMMON/COST2/RES(30),CIRKAN(30),TFIN(30),COST(30)
      EQUIVALENCE (NSET(1),QSET(1))
      NNSET=50000
      NCRDR=5
      NPRNT=6
      NTAPE=7
      CALL SLAM
      STOP
C********************************************************************
      END
      SUBROUTINE INTLC
      DIMENSION NSET(50000)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/GCOM5/ IISED(10),JJBEG,JJCLR,MMNIT,MMON,NNCFI,
     1NNDAY,NNPT,NNRNS,NNSTR,NNYR,SSEED(10),LSEED(10)
      COMMON QSET(50000)
```

```
      COMMON/SCOM9/CPROB(9),VALUE(9)
      COMMON/KANS/IOLD,INEW,NUMADD,NUMDEL
      COMMON/FACTORS/IF1,IF2,IF3,JF1,JF2,JF3,ISAMPLE
      COMMON/SAVES/SCOST,IK2,SAVG,SINV
      COMMON/SAVE2/SOINV,SOWIP,SOLEADT,SORES,SOCIR,SOWITHD,SOPROD,SOFIN
      COMMON/PPS/PPINV(30),PPWIP(30),PPLEAD(30),PPFIN(30),PPCIR(30)
      COMMON/PP2/PPRES(30),PPWITHD(30),PPPROD(30)
      COMMON/COSTS/AVGLEADA(30),AVGLEADB(30),TOTINV(30),WIPS(30)
      COMMON/COST2/RES(30),CIRKAN(30),TFIN(30),COST(30)
      EQUIVALENCE (NSET(1),QSET(1))
      REAL PINFO,X,A(6)
      INTEGER IKANS,ICNT /0/
C *******************************************************************
      ISAMPLE=1
      IIRUN=NNRUN
      NRUN=MOD(IIRUN,ISAMPLE)
      IF (NRUN.EQ.0) NRUN=ISAMPLE
      LSEED(1)=NRUN*1111
      LSEED(2)=NRUN*2222
      LSEED(3)=NRUN*3333
      LSEED(4)=NRUN*4444
      LSEED(5)=NRUN*5555
      LSEED(6)=NRUN*6666
      LSEED(7)=NRUN*7777
      LSEED(8)=NRUN*8888
      LSEED(9)=NRUN*9999
      LSEED(10)=NRUN*1010
 700  IIRUN=INT((IIRUN-1)/ISAMPLE)+1
      XX(60)=35.
      XX(61)=35.
      XX(34)=3.5
      XX(35)=3.5
      NRUN=MOD(IIRUN,700)
      IF(NRUN.EQ.0) NRUN=700
      NRUN=NRUN-1
      IOLD=INT(NRUN/70)+1
      INEW=MOD(IIRUN,10)
      IF (INEW.EQ.0) INEW=10
      NUMDEL=IOLD-INEW+2
      IF (NUMDEL.LT.2) NUMDEL=2.0
      NUMADD=INEW-IOLD
      IF (NUMADD.LT.0) NUMADD=0
      JRUN=INT((IIRUN-1)/700)+1
      IF(JRUN.GT.4) GO TO 20
      XX(20)=6.1
      XX(21)=6.1
      IF1=0
      GO TO 21
 20   XX(20)=0.1
      XX(21)=12.1
      IF1=1
 21   IF(JRUN.EQ.3.OR.JRUN.EQ.4.OR.JRUN.EQ.7.OR.JRUN.EQ.8) GO TO 22
      XX(28)=4.6
      XX(29)=0.05
      IF2=0
      GO TO 23
```

```
22  XX(28)=4.6
    XX(29)=3.0
    IF2=1
23  IF(JRUN.EQ.2.OR.JRUN.EQ.4.OR.JRUN.EQ.6.OR.JRUN.EQ.8) GO TO 24
    CPROB(1)=0.0
    CPROB(2)=0.0
    CPROB(3)=0.0
    CPROB(4)=0.33
    CPROB(5)=0.66
    CPROB(6)=1.0
    CPROB(7)=1.0
    CPROB(8)=1.0
    CPROB(9)=1.0
    IF3=0
    GO TO 25
24  CPROB(1)=.35
    CPROB(2)=.45
    CPROB(3)=.45
    CPROB(4)=.50
    CPROB(5)=.50
    CPROB(6)=.55
    CPROB(7)=.55
    CPROB(8)=.65
    CPROB(9)=1.0
    IF3=1
25  VALUE(1)=3.0
    VALUE(2)=3.5
    VALUE(3)=4.0
    VALUE(4)=4.9
    VALUE(5)=5.0
    VALUE(6)=5.1
    VALUE(7)=6.0
    VALUE(8)=6.5
    VALUE(9)=7.0
    XX(1)=0.0
    XX(2)=0.0
    XX(3)=0.0
    XX(4)=0.0
    XX(5)=0.0
    XX(6)=0.0
    XX(7)=0.0
    A(1)=NUMDEL
    CALL FILEM(69,A)
    CALL FILEM(70,A)
    CALL FILEM(71,A)
    CALL FILEM(72,A)
    CALL FILEM(73,A)
    CALL FILEM(74,A)
    CALL FILEM(75,A)
    A(1)=0.0
    DO 2 I=1,IOLD
    CALL FILEM(7,A)
    CALL FILEM(9,A)
    CALL FILEM(10,A)
    CALL FILEM(11,A)
    CALL FILEM(34,A)
```

```
      CALL FILEM(36,A)
    2 CALL FILEM(37,A)
      RETURN
C ********************************************************************
      END
      SUBROUTINE EVENT(I)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/GCOM5/ IISED(10),JJBEG,JJCLR,MMNIT,MMON,NNCFI,
     1NNDAY,NNPT,NNRNS,NNSTR,NNYR,SSEED(10),LSEED(10)
      COMMON/KANS/IOLD,INEW,NUMADD,NUMDEL
      COMMON/SCOM9/CPROB(9),VALUE(9)
      COMMON/FACTORS/IF1,IF2,IF3,JF1,JF2,JF3,ISAMPLE
      COMMON/SAVES/SCOST,IK2,SAVG,SINV
      COMMON/SAVE2/SOINV,SOWIP,SOLEADT,SORES,SOCIR,SOWITHD,SOPROD,SOFIN
      COMMON/PPS/PPINV(30),PPWIP(30),PPLEAD(30),PPFIN(30),PPCIR(30)
      COMMON/PP2/PPRES(30),PPWITHD(30),PPPROD(30)
      COMMON/COSTS/AVGLEADA(30),AVGLEADB(30),TOTINV(30),WIPS(30)
      COMMON/COST2/RES(30),CIRKAN(30),TFIN(30),COST(30)
      REAL A(6)
      GO TO (1,2),I
    1 CALL CLEAR
      RETURN
    2 XX(1)=1.0
      XX(2)=1.0
      XX(3)=1.0
      XX(4)=1.0
      XX(5)=1.0
      XX(6)=1.0
      XX(7)=1.0
      IIRUN=NNRUN
      IIRUN=INT((IIRUN-1)/ISAMPLE)+1
      ITEMP=MOD(IIRUN,70)
      IF(ITEMP.EQ.0) ITEMP=70
      ITEMP=ITEMP-1
      IRUN=INT(ITEMP/10) + 1
      IF(IIRUN.GT.700.AND.IIRUN.LE.1400) GO TO 701
      IF(IIRUN.GT.1400.AND.IIRUN.LE.2100) GO TO 141
      IF(IIRUN.GT.2100.AND.IIRUN.LE.2800) GO TO 211
      IF(IIRUN.GT.2800.AND.IIRUN.LE.3500) GO TO 281
      IF(IIRUN.GT.3500.AND.IIRUN.LE.4200) GO TO 351
      IF(IIRUN.GT.4200.AND.IIRUN.LE.4900) GO TO 421
      IF(IIRUN.GT.4900.AND.IIRUN.LE.5600) GO TO 491
      IF(IRUN.GE.4) GO TO 20
      XX(20)=6.1
      XX(21)=6.1
      JF1=0
      GO TO 21
   20  XX(20)=0.1
      XX(21)=12.1
      JF1=1
   21  IF(IRUN.EQ.2.OR.IRUN.EQ.3.OR.IRUN.EQ.6.OR.IRUN.EQ.7) GO TO 22
      XX(28)=4.6
      XX(29)=0.05
      JF2=0
      GO TO 23
```

```
22  XX(28)=4.6
    XX(29)=3.0
    JF2=1
23  IF(IRUN.EQ.1.OR.IRUN.EQ.3.OR.IRUN.EQ.5.OR.IRUN.EQ.7) GO TO 24
    CPROB(1)=0.0
    CPROB(2)=0.0
    CPROB(3)=0.0
    CPROB(4)=0.33
    CPROB(5)=0.66
    CPROB(6)=1.0
    CPROB(7)=1.0
    CPROB(8)=1.0
    CPROB(9)=1.0
    JF3=0
    GO TO 25
24  CPROB(1)=.35
    CPROB(2)=.45
    CPROB(3)=.45
    CPROB(4)=.50
    CPROB(5)=.50
    CPROB(6)=.55
    CPROB(7)=.55
    CPROB(8)=.65
    CPROB(9)=1.0
    JF3=1
    GO TO 25
701 IF(IRUN.GE.4) GO TO 30
    XX(20)=6.1
    XX(21)=6.1
    JF1=0
    GO TO 31
30  XX(20)=0.1
    XX(21)=12.1
    JF1=1
31  IF(IRUN.EQ.2.OR.IRUN.EQ.3.OR.IRUN.EQ.6.OR.IRUN.EQ.7) GO TO 32
    XX(28)=4.6
    XX(29)=0.05
    JF2=0
    GO TO 33
32  XX(28)=4.6
    XX(29)=3.0
    JF2=1
33  IF(IRUN.EQ.3.OR.IRUN.EQ.5.OR.IRUN.EQ.7) GO TO 34
    CPROB(1)=0.0
    CPROB(2)=0.0
    CPROB(3)=0.0
    CPROB(4)=0.33
    CPROB(5)=0.66
    CPROB(6)=1.0
    CPROB(7)=1.0
    CPROB(8)=1.0
    CPROB(9)=1.0
    JF3=0
    GO TO 25
34  CPROB(1)=.35
    CPROB(2)=.45
```

```
      CPROB(3)=.45
      CPROB(4)=.50
      CPROB(5)=.50
      CPROB(6)=.55
      CPROB(7)=.55
      CPROB(8)=.65
      CPROB(9)=1.0
      JF3=1
      GO TO 25
141   IF(IRUN.GE.4) GO TO 40
      XX(20)=6.1
      XX(21)=6.1
      JF1=0
      GO TO 41
 40   XX(20)=0.1
      XX(21)=12.1
      JF1=1
 41   IF(IRUN.EQ.3.OR.IRUN.EQ.6.OR.IRUN.EQ.7) GO TO 42
      XX(28)=4.6
      XX(29)=0.05
      JF2=0
      GO TO 43
 42   XX(28)=4.6
      XX(29)=3.0
      JF2=1
 43   IF(IRUN.EQ.2.OR.IRUN.EQ.3.OR.IRUN.EQ.5.OR.IRUN.EQ.7) GO TO 44
      CPROB(1)=0.0
      CPROB(2)=0.0
      CPROB(3)=0.0
      CPROB(4)=0.33
      CPROB(5)=0.66
      CPROB(6)=1.0
      CPROB(7)=1.0
      CPROB(8)=1.0
      CPROB(9)=1.0
      JF3=0
      GO TO 25
 44   CPROB(1)=.35
      CPROB(2)=.45
      CPROB(3)=.45
      CPROB(4)=.50
      CPROB(5)=.50
      CPROB(6)=.55
      CPROB(7)=.55
      CPROB(8)=.65
      CPROB(9)=1.0
      JF3=1
      GO TO 25
211   IF(IRUN.GE.4) GO TO 50
      XX(20)=6.1
      XX(21)=6.1
      JF1=0
      GO TO 51
 50   XX(20)=0.1
      XX(21)=12.1
      JF1=1
```

```
51  IF(IRUN.EQ.3.OR.IRUN.EQ.6.OR.IRUN.EQ.7) GO TO 52
    XX(28)=4.6
    XX(29)=0.05
    JF2=0
    GO TO 53
52  XX(28)=4.6
    XX(29)=3.0
    JF2=1
53  IF(IRUN.EQ.2.OR.IRUN.EQ.5.OR.IRUN.EQ.7) GO TO 54
    CPROB(1)=0.0
    CPROB(2)=0.0
    CPROB(3)=0.0
    CPROB(4)=0.33
    CPROB(5)=0.66
    CPROB(6)=1.0
    CPROB(7)=1.0
    CPROB(8)=1.0
    CPROB(9)=1.0
    JF3=0
    GO TO 25
54  CPROB(1)=.35
    CPROB(2)=.45
    CPROB(3)=.45
    CPROB(4)=.50
    CPROB(5)=.50
    CPROB(6)=.55
    CPROB(7)=.55
    CPROB(8)=.65
    CPROB(9)=1.0
    JF3=1
    GO TO 25
281 IF(IRUN.GT.4) GO TO 60
    XX(20)=6.1
    XX(21)=6.1
    JF1=0
    GO TO 61
60  XX(20)=0.1
    XX(21)=12.1
    JF1=1
61  IF(IRUN.EQ.3.OR.IRUN.EQ.4.OR.IRUN.EQ.6.OR.IRUN.EQ.7) GO TO 62
    XX(28)=4.6
    XX(29)=0.05
    JF2=0
    GO TO 63
62  XX(28)=4.6
    XX(29)=3.0
    JF2=1
63  IF(IRUN.EQ.2.OR.IRUN.EQ.4.OR.IRUN.EQ.5.OR.IRUN.EQ.7) GO TO 64
    CPROB(1)=0.0
    CPROB(2)=0.0
    CPROB(3)=0.0
    CPROB(4)=0.33
    CPROB(5)=0.66
    CPROB(6)=1.0
    CPROB(7)=1.0
    CPROB(8)=1.0
```

```
      CPROB(9)=1.0
      JF3=0
      GO TO 25
 64   CPROB(1)=.35
      CPROB(2)=.45
      CPROB(3)=.45
      CPROB(4)=.50
      CPROB(5)=.50
      CPROB(6)=.55
      CPROB(7)=.55
      CPROB(8)=.65
      CPROB(9)=1.0
      JF3=1
      GO TO 25
351   IF(IRUN.GT.4) GO TO 70
      XX(20)=6.1
      XX(21)=6.1
      JF1=0
      GO TO 71
 70   XX(20)=0.1
      XX(21)=12.1
      JF1=1
 71   IF(IRUN.EQ.3.OR.IRUN.EQ.4.OR.IRUN.EQ.6.OR.IRUN.EQ.7) GO TO 72
      XX(28)=4.6
      XX(29)=0.05
      JF2=0
      GO TO 73
 72   XX(28)=4.6
      XX(29)=3.0
      JF2=1
 73   IF(IRUN.EQ.2.OR.IRUN.EQ.4.OR.IRUN.EQ.7) GO TO 74
      CPROB(1)=0.0
      CPROB(2)=0.0
      CPROB(3)=0.0
      CPROB(4)=0.33
      CPROB(5)=0.66
      CPROB(6)=1.0
      CPROB(7)=1.0
      CPROB(8)=1.0
      CPROB(9)=1.0
      JF3=0
      GO TO 25
 74   CPROB(1)=.35
      CPROB(2)=.45
      CPROB(3)=.45
      CPROB(4)=.50
      CPROB(5)=.50
      CPROB(6)=.55
      CPROB(7)=.55
      CPROB(8)=.65
      CPROB(9)=1.0
      JF3=1
      GO TO 25
421   IF(IRUN.GT.4) GO TO 80
      XX(20)=6.1
      XX(21)=6.1
```

```
      JF1=0
      GO TO 81
 80   XX(20)=0.1
      XX(21)=12.1
      JF1=1
 81   IF(IRUN.EQ.3.OR.IRUN.EQ.4.OR.IRUN.EQ.7) GO TO 82
      XX(28)=4.6
      XX(29)=0.05
      JF2=0
      GO TO 83
 82   XX(28)=4.6
      XX(29)=3.0
      JF2=1
 83   IF(IRUN.EQ.2.OR.IRUN.EQ.4.OR.IRUN.EQ.6.OR.IRUN.EQ.7) GO TO 84
      CPROB(1)=0.0
      CPROB(2)=0.0
      CPROB(3)=0.0
      CPROB(4)=0.33
      CPROB(5)=0.66
      CPROB(6)=1.0
      CPROB(7)=1.0
      CPROB(8)=1.0
      CPROB(9)=1.0
      JF3=0
      GO TO 25
 84   CPROB(1)=.35
      CPROB(2)=.45
      CPROB(3)=.45
      CPROB(4)=.50
      CPROB(5)=.50
      CPROB(6)=.55
      CPROB(7)=.55
      CPROB(8)=.65
      CPROB(9)=1.0
      JF3=1
      GO TO 25
491   IF(IRUN.GT.4) GO TO 90
      XX(20)=6.1
      XX(21)=6.1
      JF1=0
      GO TO 91
 90   XX(20)=0.1
      XX(21)=12.1
      JF1=1
 91   IF(IRUN.EQ.3.OR.IRUN.EQ.4.OR.IRUN.EQ.7) GO TO 92
      XX(28)=4.6
      XX(29)=0.05
      JF2=0
      GO TO 93
 92   XX(28)=4.6
      XX(29)=3.0
      JF2=1
 93   IF(IRUN.EQ.2.OR.IRUN.EQ.4.OR.IRUN.EQ.6) GO TO 94
      CPROB(1)=0.0
      CPROB(2)=0.0
      CPROB(3)=0.0
```

```
      CPROB(4)=0.33
      CPROB(5)=0.66
      CPROB(6)=1.0
      CPROB(7)=1.0
      CPROB(8)=1.0
      CPROB(9)=1.0
      JF3=0
      GO TO 25
94    CPROB(1)=.35
      CPROB(2)=.45
      CPROB(3)=.45
      CPROB(4)=.50
      CPROB(5)=.50
      CPROB(6)=.55
      CPROB(7)=.55
      CPROB(8)=.65
      CPROB(9)=1.0
      JF3=1
25    VALUE(1)=3.0
      VALUE(2)=3.5
      VALUE(3)=4.0
      VALUE(4)=4.9
      VALUE(5)=5.0
      VALUE(6)=5.1
      VALUE(7)=6.0
      VALUE(8)=6.5
      VALUE(9)=7.0
437   IIRUN=NNRUN
      NRUN=MOD(IIRUN,ISAMPLE)
      IF (NRUN.EQ.0) NRUN=ISAMPLE
      PPINV(NRUN)=(CCNUM(7)*FFAWT(7))+(CCNUM(9)*FFAWT(34))
      A43=((CCNUM(17)*FFAWT(9))+(CCNUM(19)*FFAWT(10))+
     1(CCNUM(21)*FFAWT(11)))/3
      A42=((CCNUM(22)*FFAWT(36))+(CCNUM(23)*FFAWT(37)))/2
      PPWIP(NRUN)=((CCNUM(6)*CCAVG(6))+A43)+((CCNUM(16)*CCAVG(16))
     1+A42)
      PPLEAD(NRUN)=(CCAVG(7)+CCAVG(9))/2
      PPFIN(NRUN)=TNOW
      PPCIR(NRUN)=CCNUM(6)+CCNUM(16)
      PPRES(NRUN)=RRAVG(1)
      PPWITHD(NRUN)=(FFAWT(6)+FFAWT(33))/2
      PPPROD(NRUN)=(FFAWT(8)+FFAWT(35))/2
      CALL CLEAR
   87 CONTINUE
      DO 67 I = 1,NUMADD
      A(1)=0.0
      A(2)=0.0
      A(3)=0.0
      A(4)=0.0
      CALL FILEM(8,A)
      CALL FILEM(13,A)
      CALL FILEM(18,A)
      CALL FILEM(23,A)
      CALL FILEM(35,A)
      CALL FILEM(39,A)
67    CALL FILEM(44,A)
```

```
      RETURN
C *********************************************************************
      END
      FUNCTION USERF(IFN)
      COMMON/SCOM9/CPROB(9),VALUE(9)
      USERF=DPROB(CPROB,VALUE,9,IFN)
      RETURN
      END
      SUBROUTINE OTPUT
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/GCOM5/ IISED(10),JJBEG,JJCLR,MMNIT,MMON,NNCFI,
     1NNDAY,NNPT,NNRNS,NNSTR,NNYR,SSEED(10),LSEED(10)
      COMMON/KANS/IOLD,INEW,NUMADD,NUMDEL
      COMMON/SCOM9/CPROB(9),VALUE(9)
      COMMON/FACTORS/IF1,IF2,IF3,JF1,JF2,JF3,ISAMPLE
      COMMON/SAVES/SCOST,IK2,SAVG,SINV
      COMMON/SAVE2/SOINV,SOWIP,SOLEADT,SORES,SOCIR,SOWITHD,SOPROD,SOFIN
      COMMON/PPS/PPINV(30),PPWIP(30),PPLEAD(30),PPFIN(30),PPCIR(30)
      COMMON/PP2/PPRES(30),PPWITHD(30),PPPROD(30)
      COMMON/COSTS/AVGLEADA(30),AVGLEADB(30),TOTINV(30),WIPS(30)
      COMMON/COST2/RES(30),CIRKAN(30),TFIN(30),COST(30)
C-----------
      IIRUN=NNRUN
      NRUN=MOD(IIRUN,ISAMPLE)
      IF (NRUN.EQ.0) NRUN=ISAMPLE
C********************
C THIS IS THE COST CALCULATION PORTION
C
  503 AVGLEADA(NRUN)=CCAVG(7)
      AVGLEADB(NRUN)=CCAVG(9)
      TOTINV(NRUN)=(CCNUM(7)*FFAWT(7))+(CCNUM(9)*FFAWT(34))
      A43=((CCNUM(17)*FFAWT(9))+(CCNUM(19)*FFAWT(10))+
     1(CCNUM(21)*FFAWT(11)))/3
      A42=((CCNUM(22)*FFAWT(36))+(CCNUM(23)*FFAWT(37)))/2
      WIPS(NRUN)=((CCNUM(6)*CCAVG(6))+A43)+((CCNUM(16)*CCAVG(16))
     1+A42)
      RES(NRUN)=RRAVG(1)
      CIRKAN(NRUN)=CCNUM(6)+CCNUM(16)
      TFIN(NRUN)=TNOW
      COST(NRUN)=(TFIN(NRUN)+(TOTINV(NRUN)/200.)+(WIPS(NRUN)/300.))/10.
  63  IF (NRUN.NE.ISAMPLE) GO TO 87
C
C **** When NRUN = Sample size compute cost
C
      OSINV=0.
      OSWIP=0.
      OSLEAD=0.
      OSFIN=0.
      OSCIR=0.
      OSRES=0.
      OSWITHD=0.
      OSPROD=0.
      SUMINV=0.
      SUMWIP=0.
      SUMLEAD=0.
```

```
      SUMFIN=0.
      SUMCIR=0.
      SUMRES=0.
      SUMCOST=0.
C
C  IF THIS IS THE LAST RUN OF THE SAMPLE COMPUTE TRIMED MEAN
C
C
C  COMPUTE RUNNING TOTALS FOR ALL FACTORS
C
      DO 509 I=1,ISAMPLE
      SUMINV=SUMINV+TOTINV(I)
      SUMWIP=SUMWIP+WIPS(I)
      SUMLEAD=SUMLEAD+((AVGLEADA(I)+AVGLEADB(I))/2)
      SUMFIN=SUMFIN+TFIN(I)
      SUMCIR=SUMCIR+CIRKAN(I)
      SUMRES=SUMRES+RES(I)
      SUMCOST=SUMCOST+COST(I)
      OSLEAD=OSLEAD+PPLEAD(I)
      OSWITHD=OSWITHD+PPWITHD(I)
      OSPROD=OSPROD+PPPROD(I)
      OSINV=OSINV+PPINV(I)
      OSWIP=OSWIP+PPWIP(I)
      OSRES=OSRES+PPRES(I)
      OSCIR=OSCIR+PPCIR(I)
      OSFIN=OSFIN+PPFIN(I)
 509  CONTINUE
C
C  COMPUTE AVERAGES AND COST FOR ALL FACTORS OVER ISAMPLE RUNS
C
C
C  WRITE THE SUMMARY FIGURES FOR PERIOD T - 1
C
      AVGINV=SUMINV/ISAMPLE
      AVGWIP=SUMWIP/ISAMPLE
      AVGLEAD=SUMLEAD/ISAMPLE
      AVGFIN=SUMFIN/ISAMPLE
      AVGCIR=SUMCIR/ISAMPLE
      AVGRES=SUMRES/ISAMPLE
      AVGCOST=SUMCOST/ISAMPLE
      OAINV=OSINV/ISAMPLE
      OAWIP=OSWIP/ISAMPLE
      OALEADT=OSLEAD/ISAMPLE
      OAFIN=OSFIN/ISAMPLE
      OACIR=OSCIR/ISAMPLE
      OARES=OSRES/ISAMPLE
      OAWITHD=OSWITHD/ISAMPLE
      OAPROD=OSPROD/ISAMPLE
      WRITE(1,677) IF1,IF2,IF3,JF1,JF2,JF3,IOLD,INEW,OAINV,OAWIP,
     1OALEADT,OAFIN,OACIR,OARES,OAWITHD,OAPROD
 677  FORMAT(6I1,I2,I2,F7.0,F9.0,F8.1,F10.2,F7.0,F6.2,F7.2,F7.2)
C
C  WRITE FACTOR AVERAGES FOR EACH COMBINATION
C
      WRITE(1,177) IF1,IF2,IF3,JF1,JF2,JF3,IOLD,INEW,AVGINV,AVGWIP,
     1AVGLEAD,AVGFIN,AVGCIR,AVGRES,AVGCOST
```

```
 177 FORMAT(6I1,I2,I2,F7.0,F9.0,F8.1,F10.2,F7.0,F6.2,F8.2)
C
C  IF TOTCOST IS LESS THAN SCOST, SAVE THE OPTIMAL NUMBER OF KANBANS
C  AND FACTORS FROM PERIOD I-1
C
 678  IIRUN=INT((IIRUN-1)/ISAMPLE)+1
      ITIME=MOD(IIRUN,10)
      IF (ITIME.NE.1) GOTO 777
      SCOST=99999.
 777  IF(AVGCOST.GE.SCOST) GO TO 573
      SCOST=AVGCOST
      IK2=INEW
      SOCIR=OACIR
      SOINV=OAINV
      SOLEADT=OALEADT
      SOFIN=OAFIN
      SOWIP=OAWIP
      SORES=OARES
      SOWITHD=OAWITHD
      SOPROD=OAPROD
  573 IF(ITIME.NE.0) GO TO 87
      WRITE(1,173) IF1,IF2,IF3,JF1,JF2,JF3,IOLD,IK2,SOINV,SOWIP,
     1SOLEADT,SOFIN,SOCIR,SORES,SCOST,SOWITHD,SOPROD
  173 FORMAT(6I1,I2,I2,F7.0,F9.0,F8.1,F10.2,F7.0,F6.2,F8.2,2(F7.2))
   87 CONTINUE
      RETURN
C***********************************************************************
      END
//GO.SYSIN DD *
;SLAM CARDS
GEN,BAWRAY,PROBLEM,4/23/91,50,NO,NO,YES/YES,NO,NO,72;
LIMITS,75,4,400 ;
;MONTR,TRACE,999,1010;
PRIORITY/12,FIFO; USED TO BE LVF(2)
SEEDS,23444(1)/NO,33239(2)/NO,98723(3)/NO,68943(4)/NO,256777(5)/NO;
SEEDS,876215(6)/NO,87655(7)/NO,56955(8)/NO,733333(9)/NO ;
SEEDS,4433342(10)/NO ;
INIT,0.0,,Y,Y,Y;
;
; JIT shop with 6 workstations
;
NETWORK;
;
;
      RESOURCE/MACH2(2),12,5 ;
      RESOURCE/MACH3(1),17 ;
      RESOURCE/MACH4(1),22 ;
      RESOURCE/MACH5(1),27 ;
      RESOURCE/MACH7(1),43 ;
      RESOURCE/MACH8(1),48 ;
;
;  *1
;
      CREATE,UNFRM(XX(60),XX(61),1),0.;
      ACT,0 ;
      ASSIGN,ATRIB(1)=UNFRM(XX(20),XX(21),1) ;
```

```
     ACT,0 ;
     GOON,13;
     ACT,,ATRIB(1).GE.0.,C1 ;
     ACT,,ATRIB(1).GE.1.,C1 ;
     ACT,,ATRIB(1).GE.2.,C1 ;
     ACT,,ATRIB(1).GE.3.,C1 ;
     ACT,,ATRIB(1).GE.4.,C1 ;
     ACT,,ATRIB(1).GE.5.,C1 ;
     ACT,,ATRIB(1).GE.6.,C1 ;
     ACT,,ATRIB(1).GE.7.,C1 ;
     ACT,,ATRIB(1).GE.8.,C1 ;
     ACT,,ATRIB(1).GE.9.,C1 ;
     ACT,,ATRIB(1).GE.10.,C1 ;
     ACT,,ATRIB(1).GE.11.,C1 ;
     ACT,,ATRIB(1).GE.12.,C1 ;
C1   COLECT(11),INT(1),NUM DEM WC2a ;  THIS IS NUMBER DEMANDED FOR 2a
     ACT,0,,A2A;
;  *2A
;  WITHDRAW KANBAN POST FOR WORKCENTER 2
;
A2A   ASSIGN,ATRIB(4)=TNOW ;
     ACT,0 ;
Q21   QUEUE(6),,,,S21 ;
;
;    WORKCENTER 2
;
Q22   QUEUE(7),,,,S21 ;
S21   SEL,ASM,,,Q21,Q22 ;
     ACT,0 ;
     GOON,3 ;
     ACT,,0.EQ.0,DNE;
;
;  PRODUCTION  KANBAN FOR WORKCENTER 2 IS THE NEXT ACTIVITY
     ACT,,XX(1).EQ.0.0,Q23 ;
     ACT,,XX(1).EQ.1.0,Q1 ;
;
Q23   QUEUE(8),,,,S22 ;
Q24   QUEUE(9),,,,S22 ;
Q26   QUEUE(10),,,,S22 ;
Q27   QUEUE(11),,,,S22 ;
S22   SEL,ASM,,,Q23,Q24,Q26,Q27 ;
     ACT,0 ;
     GOON,4 ;
     ACT,,XX(2).EQ.0.0,Q41 ;
     ACT,,XX(2).EQ.1.0,Q2 ;
     ACT,,XX(3).EQ.0.0,Q61 ;
     ACT,,XX(3).EQ.1.0,Q3 ;
     ACT,,XX(4).EQ.0.0,Q81 ;
     ACT,,XX(4).EQ.1.0,Q4 ;
     ACT,, ;
     ASSIGN,ATRIB(3)=TNOW ;
     ACT,0 ;
     AWAIT(12/50),MACH2/1,,1 ;
     ACT/1,RLOGN(XX(28),XX(29),3) ; THIS IS PROC TIME FOR WORKCENTER 2
 F1   FREE,MACH2/1 ;
     ACT,0 ;
```

```
     COLECT(6),INT(3),WIP WC2 ;    THIS IS WIP FOR WORKCENTER 2
     ACT,0 ;
     COLECT(7),INT(4),LEAD 2A ;    THIS IS PRODUCTION LEADTIME 2A
     ACT,0,,Q22 ;
;
; *3
; WITHDRAW KANBAN POST FOR WORKCENTER 3
;
Q41  QUEUE(13),,,,S41 ;
;
;  WORKCENTER 3
;
Q42  QUEUE(14),1,,,S41 ;
S41  SEL,ASM,,,Q41,Q42 ;
     ACT,0 ;
     COLECT(17),ALL,NUM from WC3 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q24 ;
;
; PRODUCTION  KANBAN FOR WORKCENTER 3 IS THE NEXT ACTIVITY
     ACT,,,Q43 ;
;
Q43  QUEUE(15),,,,S42 ;
Q44  QUEUE(16),1,,,S42 ;
S42  SEL,ASM,,,Q43,Q44 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q45 ;
     ACT,0 ;
     ASSIGN,ATRIB(3)=TNOW ;
     ACT,0 ;
A41  AWAIT(17/50),MACH3/1,,1 ;
     ACT,UNFRM(XX(34),XX(35),10) ; PROCESSING TIME FOR WORKCENTER 3
     FREE,MACH3/1 ;
     ACT,0 ;
     COLECT(8),INT(3),WIP WC3 ;
     ACT,0,,Q42 ;
;
;
Q45  QUEUE(49),,,,S43 ;
;                   THIS IS VENDOR FOR WORKCENTER 3
;
Q46  QUEUE(50),1,,,S43 ;
S43  SEL,ASM,,,Q45,Q46 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q44 ;
     ACT,,,Q47 ;
Q47  QUEUE(51),,,,S44 ;
Q48  QUEUE(52),1,,,S44 ;
S44  SEL,ASM,,,Q47,Q48 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q48 ;
```

```
     ACT,USERF(4),,Q46 ;
;
; *4
;  WORKCENTER 4
; WITHDRAW KANBAN POST FOR WORKCENTER 4
;
Q61  QUEUE(18),,,,S61 ;
Q62  QUEUE(19),1,,,S61 ;
S61  SEL,ASM,,,Q61,Q62 ;
     ACT,0 ;
     COLECT(19),ALL,NUM from WC4 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q26 ;
;
; PRODUCTION  KANBAN FOR WORKCENTER 3 IS THE NEXT ACTIVITY
     ACT,,,Q63 ;
;
Q63  QUEUE(20),,,,S62 ;
Q64  QUEUE(21),1,,,S62 ;
S62  SEL,ASM,,,Q63,Q64 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q65 ;
     ACT,0 ;
     ASSIGN,ATRIB(3)=TNOW ;
     ACT,0 ;
A61  AWAIT(22/50),MACH4/1,,1 ;
     ACT,UNFRM(XX(34),XX(35),10) ; PROCESSING TIME FOR WORKCENTER 4
     FREE,MACH4/1 ;
     ACT,0 ;
     COLECT(10),INT(3),WIP WC4 ;   THIS IS WIP FOR WORKCENTER 4
     ACT,0,,Q62 ;
;
;
Q65  QUEUE(53),,,,S63 ;
;                   THIS IS VENDOR FOR WORKCENTER 4
;
Q66  QUEUE(54),1,,,S63 ;
S63  SEL,ASM,,,Q65,Q66 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q64 ;
     ACT,,,Q67 ;
Q67  QUEUE(55),,,,S64 ;
Q68  QUEUE(56),1,,,S64 ;
S64  SEL,ASM,,,Q67,Q68 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q68 ;
     ACT,USERF(5),,Q66 ;
;
; *5
;
; WITHDRAW KANBAN POST FOR WORKCENTER 5
;
```

```
Q81   QUEUE(23),,,,S81 ;
;
;   WORKCENTER 5
;
Q82   QUEUE(24),1,,,S81 ;
S81   SEL,ASM,,,Q81,Q82 ;
      ACT,0 ;
      COLECT(21),ALL,NUM from WC5 ;
      ACT,0 ;
      GOON,2 ;
      ACT,,,Q27 ;
;
; PRODUCTION  KANBAN FOR WORKCENTER 5 IS THE NEXT ACTIVITY
      ACT,,,Q83 ;
;
Q83   QUEUE(25),,,,S82 ;
Q84   QUEUE(26),1,,,S82 ;
S82   SEL,ASM,,,Q83,Q84 ;
      ACT,0 ;
      GOON,2 ;
      ACT,,,Q85 ;
      ACT,0 ;
      ASSIGN,ATRIB(3)=TNOW ;
      ACT,0 ;
A81   AWAIT(27/50),MACH5/1,,1 ;
      ACT,UNFRM(XX(34),XX(35),10) ; PROCESSING TIME FOR WORKCENTER 5
      FREE,MACH5/1 ;
      ACT,0 ;
      COLECT(12),INT(3),WIP WC5 ;    THIS IS WIP FOR WORKCENTER 5
      ACT,0,,Q82 ;
;
;
;
;
Q85   QUEUE(57),,,,S83 ;
;                     THIS IS VENDOR FOR WORKCENTER 5
;
Q86   QUEUE(58),1,,,S83 ;
S83   SEL,ASM,,,Q85,Q86 ;
      ACT,0 ;
      GOON,2 ;
      ACT,,,Q84 ;
      ACT,,,Q87 ;
Q87   QUEUE(59),,,,S84 ;
Q88   QUEUE(60),1,,,S84 ;
S84   SEL,ASM,,,Q87,Q88 ;
      ACT,0 ;
      GOON,2 ;
      ACT,,,Q88 ;
      ACT,USERF(6),,Q86 ;
;
; *6
;
; WORKCENTER 6
;
      CREATE,UNFRM(XX(60),XX(61),1),0;
```

```
     ACT,0 ;
     ASSIGN,ATRIB(1)=UNFRM(XX(20),XX(21),1) ;
     ACT,0 ;
     GOON,13 ;
     ACT,,ATRIB(1).GE.0.,C2 ;
     ACT,,ATRIB(1).GE.1.,C2 ;
     ACT,,ATRIB(1).GE.2.,C2 ;
     ACT,,ATRIB(1).GE.3.,C2 ;
     ACT,,ATRIB(1).GE.4.,C2 ;
     ACT,,ATRIB(1).GE.5.,C2 ;
     ACT,,ATRIB(1).GE.6.,C2 ;
     ACT,,ATRIB(1).GE.7.,C2 ;
     ACT,,ATRIB(1).GE.8.,C2 ;
     ACT,,ATRIB(1).GE.9.,C2 ;
     ACT,,ATRIB(1).GE.10.,C2 ;
     ACT,,ATRIB(1).GE.11.,C2 ;
     ACT,,ATRIB(1).GE.12.,C2 ;
C2   COLECT(13),INT(1),NUM DEM2B ;   THIS IS NUMBER DEMANDED FOR2b
     ACT,0,,A2B ;
;
; *2B
; WITHDRAW KANBAN POST FOR WORKCENTER 2B
;
A2B   ASSIGN,ATRIB(4)=TNOW ;
     ACT,0 ;
Q121  QUEUE(33),,,,S121 ;
;
;   WORKCENTER 2
;
Q122  QUEUE(34),,,,S121 ;
S121  SEL,ASM,,,Q121,Q122 ;
     ACT,0 ;
     GOON,3 ;
     ACT,,0.EQ.0,DNE2 ;
;
; PRODUCTION  KANBAN FOR WORKCENTER 2 IS THE NEXT ACTIVITY
     ACT,,XX(5).EQ.0.0,Q123 ;
     ACT,,XX(5).EQ.1.0,Q5 ;
;
Q123  QUEUE(35),,,,S122 ;
Q124  QUEUE(36),,,,S122 ;
Q126  QUEUE(37),,,,S122 ;
S122  SEL,ASM,,,Q123,Q124,Q126 ;
     ACT,0 ;
     GOON,3 ;
     ACT,,XX(6).EQ.0.0,Q141 ;
     ACT,,XX(6).EQ.1.0,Q6 ;
     ACT,,XX(7).EQ.0.0,Q161 ;
     ACT,,XX(7).EQ.1.0,Q7 ;
     ACT,, ;
     ASSIGN,ATRIB(3)=TNOW ;
     ACT,0 ;
     AWAIT(5/50),MACH2/1,,1 ;
     ACT/2,RLOGN(XX(28),XX(29),9) ; PROC TIME FOR WORKCENTER 2B
 F2   FREE,MACH2/1 ;
     ACT,0 ;
```

```
    COLECT(16),INT(3),WIP WC2B ;    THIS IS WIP FOR WORKCENTER 2
    ACT,0 ;
    COLECT(9),INT(4),LEAD 2B ;    PRODUCTION LEADTIME 2B
    ACT,0,,Q122 ;
;
;  *7
;  WITHDRAW KANBAN POST FOR WORKCENTER 7
;
Q141  QUEUE(39),,,,S141 ;
;
;   WORKCENTER 7
;
Q142  QUEUE(40),1,,,S141 ;
S141  SEL,ASM/HIGH(2),,,Q141,Q142 ;
    ACT,0 ;
    COLECT(22),ALL,NUM from WC7 ;
    ACT,0 ;
    GOON,2 ;
    ACT,,,Q124 ;
;
;  PRODUCTION  KANBAN FOR WORKCENTER 7 IS THE NEXT ACTIVITY
    ACT,,,Q143 ;
;
Q143  QUEUE(41),,,,S142 ;
Q144  QUEUE(42),1,,,S142 ;
S142  SEL,ASM,,,Q143,Q144 ;
    ACT,0 ;
    GOON,2 ;
    ACT,,,Q145 ;
    ACT,0 ;
    ASSIGN,ATRIB(3)=TNOW ;
    ACT,0 ;
A141  AWAIT(43/50),MACH7/1,,1 ;
    ACT,UNFRM(XX(34),XX(35),10) ; PROCESSING TIME FOR WORKCENTER 7
    FREE,MACH7/1 ;
    ACT,0 ;
    COLECT(18),INT(3),WIP WC7 ;    THIS IS WIP FOR WORKCENTER 7
    ACT,0,,Q142 ;
;
;
Q145  QUEUE(61),,,,S143 ;
;                       THIS IS VENDOR FOR WORKCENTER 7
;
Q146   QUEUE(62),1,,,S143 ;
S143   SEL,ASM,,,Q145,Q146 ;
    ACT,0 ;
    GOON,2 ;
    ACT,,,Q144 ;
    ACT,,,Q147 ;
Q147   QUEUE(63),,,,S144 ;
Q148   QUEUE(64),1,,,S144 ;
S144   SEL,ASM,,,Q147,Q148 ;
    ACT,0 ;
    GOON,2 ;
    ACT,,,Q148 ;
```

```
     ACT,USERF(2),,Q146 ;
;
;
; *8
;
; WITHDRAW KANBAN POST FOR WORKCENTER 8
;
Q161  QUEUE(44),,,,S161 ;
;
;   WORKCENTER 8
;
Q162  QUEUE(45),1,,,S161 ;
S161  SEL,ASM,,,Q161,Q162 ;
     ACT,0 ;
     COLECT(23),ALL,NUM from WC8 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q126 ;
;
; PRODUCTION  KANBAN FOR WORKCENTER 8 IS THE NEXT ACTIVITY
     ACT,,,Q163 ;
;
Q163  QUEUE(46),,,,S162 ;
Q164  QUEUE(47),1,,,S162 ;
S162  SEL,ASM,,,Q163,Q164 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q165 ;
     ACT,0 ;
     ASSIGN,ATRIB(3)=TNOW ;
     ACT,0 ;
A161  AWAIT(48/50),MACH8/1,,1 ;
     ACT,UNFRM(XX(34),XX(35),10) ; PROCESSING TIME FOR WORKCENTER 8
     FREE,MACH8/1 ;
     ACT,0 ;
     COLECT(20),INT(3),WIP WC8 ;    THIS IS WIP FOR WORKCENTER 8
     ACT,0,,Q162 ;
;
;
Q165  QUEUE(65),,,,S163 ;
;                     THIS IS VENDOR FOR WORKCENTER 8
;
Q166  QUEUE(66),1,,,S163 ;
S163  SEL,ASM,,,Q165,Q166 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q164 ;
     ACT,,,Q167 ;
Q167   QUEUE(67),,,,S164 ;
Q168   QUEUE(68),1,,,S164 ;
S164   SEL,ASM,,,Q167,Q168 ;
     ACT,0 ;
     GOON,2 ;
     ACT,,,Q168 ;
     ACT,USERF(7),,Q166 ;
;
```

```
 Q1   QUEUE(69),,,, ;
      ACT,,, ;
      ACCU,ATRIB(1),,,;
      ACT,,, ;
      ASSIGN,XX(1)=0.0 ;
      ACT,,,Q23 ;
;
 Q2   QUEUE(70),,,, ;
      ACT,,, ;
      ACCU,ATRIB(1),,,;
      ACT,,, ;
      ASSIGN,XX(2)=0.0 ;
      ACT,,,Q41;
;
 Q3   QUEUE(71),,,, ;
      ACT,,, ;
      ACCU,ATRIB(1),,,;
      ACT,,, ;
      ASSIGN,XX(3)=0.0 ;
      ACT,,,Q61 ;
;
 Q4   QUEUE(72),,,, ;
      ACT,,, ;
      ACCU,ATRIB(1),,,;
      ACT,,, ;
      ASSIGN,XX(4)=0.0 ;
      ACT,,,Q81 ;
;
 Q5   QUEUE(73),,,, ;
      ACT,,, ;
      ACCU,ATRIB(1),,,;
      ACT,,, ;
      ASSIGN,XX(5)=0.0 ;
      ACT,,,Q123 ;
;
 Q6   QUEUE(74),,,, ;
      ACT,,, ;
      ACCU,ATRIB(1),,,;
      ACT,,, ;
      ASSIGN,XX(6)=0.0 ;
      ACT,,,Q141 ;
;
 Q7   QUEUE(75),,,, ;
      ACT,,, ;
      ACCU,ATRIB(1),,,;
      ACT,,,;
      ASSIGN,XX(7)=0.0 ;
      ACT,,,Q161 ;
;
DNE2  COLECT(2),INT(1),PART 2 TIS,,1;
      ACT,0,,TM ;
;
DNE   COLECT(1),INT(1),PART 1 TIS,,1;
      ACT,0,,TM ;
;
TM    GOON,3 ;
```

```
     ACT,,,NDY ;
     ACT,,,KIL ;
     ACT,0 ;
     ACCUM,300,9999 ;
     ACT,0 ;
     EVENT,1 ;
     ACT,0 ;
     TERMINATE ;
NDY  ACCUM,400,9999 ;
     ACT,0 ;
     EVENT,2 ;
     ACT,0 ;
     TERMINATE ;
KIL  TERMINATE,550;
     END;
FIN;
SIMULATE;
/*
//GO.FT01F001 DD SYSOUT=*
//GO.FT02F001 DD SYSOUT=*
/*
```

Appendix C
TR280.NNA - Training Data for the Neural Network

0 0 0 0 0 1	1	39.00	949.00	18.30	993.85	0.10	13.10	3
0 0 0 0 1 0	1	39.00	949.00	18.30	993.85	0.10	13.10	4
0 0 0 0 0 1	2	238.00	1473.00	18.20	989.23	0.10	10.29	2
0 0 0 1 0 0	2	238.00	1473.00	18.20	989.23	0.10	10.29	5
0 0 0 0 0 1	3	504.00	1956.00	18.30	984.63	0.10	8.24	1
0 0 0 0 1 0	4	731.00	2525.00	18.40	980.00	0.10	5.66	2
0 0 0 0 1 1	4	731.00	2525.00	18.40	980.00	0.10	5.66	2
0 0 0 1 0 0	4	731.00	2525.00	18.40	980.00	0.10	5.66	3
0 0 0 1 0 1	4	731.00	2525.00	18.40	980.00	0.10	5.66	3
0 0 0 1 1 0	4	731.00	2525.00	18.40	980.00	0.10	5.66	2
0 0 0 0 1 0	5	984.00	3058.00	18.40	980.00	0.10	3.47	2
0 0 0 0 1 1	5	984.00	3058.00	18.40	980.00	0.10	3.47	2
0 0 0 1 0 0	5	984.00	3058.00	18.40	980.00	0.10	3.47	3
0 0 0 1 1 0	5	984.00	3058.00	18.40	980.00	0.10	3.47	3
0 0 0 0 1 0	6	1264.00	3501.00	18.40	980.00	0.10	1.59	2
0 0 0 1 0 0	6	1264.00	3501.00	18.40	980.00	0.10	1.59	2
0 0 0 0 0 1	7	1550.00	3928.00	18.40	980.00	0.10	0.00	1
0 0 0 0 1 0	7	1550.00	3928.00	18.40	980.00	0.10	0.00	2
0 0 0 1 0 1	7	1550.00	3928.00	18.40	980.00	0.10	0.00	3
0 0 0 1 1 0	7	1550.00	3928.00	18.40	980.00	0.10	0.00	3
0 0 0 0 1 0	8	1966.00	4326.00	18.40	980.00	0.10	0.00	2
0 0 0 0 1 1	8	1966.00	4326.00	18.40	980.00	0.10	0.00	2
0 0 0 1 0 0	8	1966.00	4326.00	18.40	980.00	0.10	0.00	3
0 0 0 1 1 1	8	1966.00	4326.00	18.40	980.00	0.10	0.00	4
0 0 0 0 0 1	9	2367.00	4710.00	18.40	980.00	0.10	0.00	2
0 0 0 0 1 1	9	2367.00	4710.00	18.40	980.00	0.10	0.00	2
0 0 0 1 0 0	9	2367.00	4710.00	18.40	980.00	0.10	0.00	2
0 0 0 1 1 0	9	2367.00	4710.00	18.40	980.00	0.10	0.00	3
0 0 0 0 1 0	10	2754.00	5081.00	18.40	980.00	0.10	0.00	2
0 0 0 0 1 1	10	2754.00	5081.00	18.40	980.00	0.10	0.00	2
0 0 0 1 0 0	10	2754.00	5081.00	18.40	980.00	0.10	0.00	3
0 0 0 1 0 1	10	2754.00	5081.00	18.40	980.00	0.10	0.00	4
0 0 0 1 1 0	10	2754.00	5081.00	18.40	980.00	0.10	0.00	3
0 0 1 0 1 1	1	6.00	804.00	35.10	1014.09	0.10	26.65	2
0 0 1 1 1 0	1	6.00	804.00	35.10	1014.09	0.10	26.65	2
0 0 1 0 0 0	2	70.00	1081.00	25.90	997.48	0.10	15.73	1
0 0 1 0 1 0	2	70.00	1081.00	25.90	997.48	0.10	15.73	2
0 0 1 0 1 1	2	70.00	1081.00	25.90	997.48	0.10	15.73	2
0 0 1 1 0 0	2	70.00	1081.00	25.90	997.48	0.10	15.73	3

0	0	1	1	0	1	2	70.00	1081.00	25.90	997.48	0.10	15.73	4
0	0	1	1	1	1	2	70.00	1081.00	25.90	997.48	0.10	15.73	4
0	0	1	0	1	0	3	271.00	1398.00	21.90	989.35	0.10	9.48	2
0	0	1	1	0	1	3	271.00	1398.00	21.90	989.35	0.10	9.48	5
0	0	1	1	1	0	3	271.00	1398.00	21.90	989.35	0.10	9.48	4
0	0	1	1	1	1	3	271.00	1398.00	21.90	989.35	0.10	9.48	5
0	0	1	0	0	0	4	572.00	1761.00	19.40	982.21	0.10	5.15	1
0	0	1	0	1	1	4	572.00	1761.00	19.40	982.21	0.10	5.15	2
0	0	1	1	0	0	4	572.00	1761.00	19.40	982.21	0.10	5.15	3
0	0	1	0	0	0	5	905.00	2181.00	18.40	980.00	0.10	2.78	1
0	0	1	0	1	1	5	905.00	2181.00	18.40	980.00	0.10	2.78	2
0	0	1	1	0	0	5	905.00	2181.00	18.40	980.00	0.10	2.78	3
0	0	1	0	0	0	6	1234.00	2654.00	18.40	980.00	0.10	1.35	1
0	0	1	0	1	0	6	1234.00	2654.00	18.40	980.00	0.10	1.35	2
0	0	1	1	0	0	6	1234.00	2654.00	18.40	980.00	0.10	1.35	3
0	0	1	1	0	1	6	1234.00	2654.00	18.40	980.00	0.10	1.35	4
0	0	1	1	1	0	6	1234.00	2654.00	18.40	980.00	0.10	1.35	3
0	0	1	1	1	1	6	1234.00	2654.00	18.40	980.00	0.10	1.35	4
0	0	1	0	0	0	7	1547.00	3108.00	18.40	980.00	0.10	0.00	1
0	0	1	0	1	1	7	1547.00	3108.00	18.40	980.00	0.10	0.00	2
0	0	1	1	0	0	7	1547.00	3108.00	18.40	980.00	0.10	0.00	2
0	0	1	1	0	1	7	1547.00	3108.00	18.40	980.00	0.10	0.00	2
0	0	1	0	0	0	8	1966.00	3541.00	18.40	980.00	0.10	0.00	1
0	0	1	0	1	0	8	1966.00	3541.00	18.40	980.00	0.10	0.00	2
0	0	1	1	0	1	8	1966.00	3541.00	18.40	980.00	0.10	0.00	3
0	0	1	1	1	0	8	1966.00	3541.00	18.40	980.00	0.10	0.00	3
0	0	1	1	1	1	8	1966.00	3541.00	18.40	980.00	0.10	0.00	3
0	0	1	1	1	0	9	2367.00	3956.00	18.40	980.00	0.10	0.00	3
0	0	1	0	0	0	10	2754.00	4351.00	18.40	980.00	0.10	0.00	1
0	0	1	0	1	0	10	2754.00	4351.00	18.40	980.00	0.10	0.00	2
0	0	1	1	0	1	10	2754.00	4351.00	18.40	980.00	0.10	0.00	4
0	0	1	1	1	1	10	2754.00	4351.00	18.40	980.00	0.10	0.00	4
0	1	0	0	1	1	1	0.00	725.00	51.60	1030.50	0.10	38.91	4
0	1	0	1	0	0	1	0.00	725.00	51.60	1030.50	0.10	38.91	3
0	1	0	1	0	1	1	0.00	725.00	51.60	1030.50	0.10	38.91	2
0	1	0	1	1	0	1	0.00	725.00	51.60	1030.50	0.10	38.91	4
0	1	0	0	1	1	2	167.00	1149.00	28.10	1000.55	0.10	16.97	5
0	1	0	1	0	0	2	167.00	1149.00	28.10	1000.55	0.10	16.97	5
0	1	0	1	0	1	2	167.00	1149.00	28.10	1000.55	0.10	16.97	6
0	1	0	0	0	0	3	428.00	1542.00	25.50	993.56	0.10	13.36	1
0	1	0	1	0	0	3	428.00	1542.00	25.50	993.56	0.10	13.36	5
0	1	0	1	1	0	3	428.00	1542.00	25.50	993.56	0.10	13.36	4
0	1	0	1	1	1	3	428.00	1542.00	25.50	993.56	0.10	13.36	6
0	1	0	0	0	0	4	648.00	2103.00	23.70	985.85	0.10	9.42	1
0	1	0	0	1	1	4	648.00	2103.00	23.70	985.85	0.10	9.42	2
0	1	0	1	0	0	4	648.00	2103.00	23.70	985.85	0.10	9.42	3
0	1	0	1	1	1	4	648.00	2103.00	23.70	985.85	0.10	9.42	4
0	1	0	0	0	0	5	884.00	2734.00	23.10	981.31	0.10	6.50	1
0	1	0	0	1	1	5	884.00	2734.00	23.10	981.31	0.10	6.50	2
0	1	0	1	0	1	5	884.00	2734.00	23.10	981.31	0.10	6.50	3
0	1	0	1	1	0	5	884.00	2734.00	23.10	981.31	0.10	6.50	3
0	1	0	0	0	1	6	1141.00	3329.00	22.50	980.00	0.10	4.16	1
0	1	0	1	1	0	6	1141.00	3329.00	22.50	980.00	0.10	4.16	3
0	1	0	0	0	0	7	1416.00	3861.00	21.90	980.00	0.10	1.89	1
0	1	0	0	1	1	7	1416.00	3861.00	21.90	980.00	0.10	1.89	2
0	1	0	0	0	0	8	1777.00	4349.00	21.80	980.00	0.10	1.09	1

0	1	0	1	0	0	8	1777.00	4349.00	21.80	980.00	0.10	1.09	3
0	1	0	1	0	1	8	1777.00	4349.00	21.80	980.00	0.10	1.09	3
0	1	0	0	0	0	9	2154.00	4797.00	21.80	980.00	0.10	0.69	1
0	1	0	0	0	1	9	2154.00	4797.00	21.80	980.00	0.10	0.69	2
0	1	0	0	1	1	9	2154.00	4797.00	21.80	980.00	0.10	0.69	2
0	1	0	1	0	0	9	2154.00	4797.00	21.80	980.00	0.10	0.69	2
0	1	0	1	1	1	9	2154.00	4797.00	21.80	980.00	0.10	0.69	4
0	1	0	0	0	1	10	2532.00	5229.00	21.80	980.00	0.10	0.44	2
0	1	0	1	0	0	10	2532.00	5229.00	21.80	980.00	0.10	0.44	3
0	1	0	1	1	0	10	2532.00	5229.00	21.80	980.00	0.10	0.44	3
0	1	0	1	1	1	10	2532.00	5229.00	21.80	980.00	0.10	0.44	4
0	1	1	0	0	1	1	0.00	811.00	83.80	1067.16	0.09	62.53	1
0	1	1	0	1	0	1	0.00	811.00	83.80	1067.16	0.09	62.53	4
0	1	1	1	0	0	1	0.00	811.00	83.80	1067.16	0.09	62.53	1
0	1	1	1	1	0	1	0.00	811.00	83.80	1067.16	0.09	62.53	4
0	1	1	1	0	1	2	40.00	1176.00	37.10	1009.25	0.10	24.68	4
0	1	1	1	1	0	2	40.00	1176.00	37.10	1009.25	0.10	24.68	3
0	1	1	0	0	0	3	182.00	1485.00	30.80	997.69	0.10	16.15	1
0	1	1	0	0	1	3	182.00	1485.00	30.80	997.69	0.10	16.15	2
0	1	1	0	1	0	3	182.00	1485.00	30.80	997.69	0.10	16.15	2
0	1	1	1	0	1	3	182.00	1485.00	30.80	997.69	0.10	16.15	6
0	1	1	1	1	0	3	182.00	1485.00	30.80	997.69	0.10	16.15	4
0	1	1	0	1	0	4	428.00	1876.00	27.30	987.20	0.10	10.77	2
0	1	1	1	0	0	4	428.00	1876.00	27.30	987.20	0.10	10.77	3
0	1	1	1	0	1	4	428.00	1876.00	27.30	987.20	0.10	10.77	6
0	1	1	1	1	1	4	428.00	1876.00	27.30	987.20	0.10	10.77	5
0	1	1	0	0	0	5	744.00	2260.00	24.10	981.23	0.10	6.35	1
0	1	1	0	0	1	5	744.00	2260.00	24.10	981.23	0.10	6.35	1
0	1	1	1	1	0	5	744.00	2260.00	24.10	981.23	0.10	6.35	3
0	1	1	0	0	0	6	1110.00	2687.00	22.10	980.00	0.10	3.62	1
0	1	1	1	1	1	6	1110.00	2687.00	22.10	980.00	0.10	3.62	4
0	1	1	0	0	0	7	1414.00	3220.00	21.80	980.00	0.10	1.81	1
0	1	1	0	1	0	7	1414.00	3220.00	21.80	980.00	0.10	1.81	2
0	1	1	1	0	0	7	1414.00	3220.00	21.80	980.00	0.10	1.81	2
0	1	1	1	0	1	7	1414.00	3220.00	21.80	980.00	0.10	1.81	2
0	1	1	1	1	1	7	1414.00	3220.00	21.80	980.00	0.10	1.81	4
0	1	1	0	0	0	8	1780.00	3742.00	21.70	980.00	0.10	1.08	1
0	1	1	1	0	0	8	1780.00	3742.00	21.70	980.00	0.10	1.08	3
0	1	1	0	1	0	9	2154.00	4217.00	21.80	980.00	0.10	0.69	2
0	1	1	1	1	1	9	2154.00	4217.00	21.80	980.00	0.10	0.69	4
0	1	1	0	0	1	10	2532.00	4672.00	21.80	980.00	0.10	0.44	2
0	1	1	1	0	0	10	2532.00	4672.00	21.80	980.00	0.10	0.44	3
0	1	1	1	1	0	10	2532.00	4672.00	21.80	980.00	0.10	0.44	3
1	0	0	0	0	0	1	44.00	630.00	83.30	1118.48	0.09	55.58	1
1	0	0	0	0	1	1	44.00	630.00	83.30	1118.48	0.09	55.58	2
1	0	0	1	0	1	1	44.00	630.00	83.30	1118.48	0.09	55.58	2
1	0	0	1	1	1	2	123.00	748.00	79.20	1109.40	0.09	50.57	2
1	0	0	0	0	1	3	235.00	914.00	75.50	1102.33	0.09	44.72	2
1	0	0	1	0	1	3	235.00	914.00	75.50	1102.33	0.09	44.72	2
1	0	0	1	1	1	3	235.00	914.00	75.50	1102.33	0.09	44.72	2
1	0	0	0	0	0	4	386.00	1084.00	71.80	1095.26	0.09	38.90	1
1	0	0	0	1	1	4	386.00	1084.00	71.80	1095.26	0.09	38.90	2
1	0	0	1	0	1	4	386.00	1084.00	71.80	1095.26	0.09	38.90	3
1	0	0	1	1	0	4	386.00	1084.00	71.80	1095.26	0.09	38.90	4
1	0	0	0	0	0	5	571.00	1270.00	68.80	1090.47	0.09	36.64	6
1	0	0	0	1	1	5	571.00	1270.00	68.80	1090.47	0.09	36.64	2

1	0	0	1	0	1	5	571.00	1270.00	68.80	1090.47	0.09	36.64	3
1	0	0	1	1	0	5	571.00	1270.00	68.80	1090.47	0.09	36.64	5
1	0	0	1	1	1	5	571.00	1270.00	68.80	1090.47	0.09	36.64	5
1	0	0	0	0	1	6	821.00	1502.00	65.20	1085.53	0.09	31.71	2
1	0	0	0	1	0	6	821.00	1502.00	65.20	1085.53	0.09	31.71	2
1	0	0	0	1	1	6	821.00	1502.00	65.20	1085.53	0.09	31.71	2
1	0	0	1	1	0	6	821.00	1502.00	65.20	1085.53	0.09	31.71	6
1	0	0	1	1	1	6	821.00	1502.00	65.20	1085.53	0.09	31.71	7
1	0	0	0	1	0	7	1150.00	1768.00	62.20	1080.29	0.10	28.10	2
1	0	0	1	1	0	7	1150.00	1768.00	62.20	1080.29	0.10	28.10	7
1	0	0	0	0	0	8	1459.00	2006.00	59.20	1073.36	0.10	24.53	1
1	0	0	0	1	1	8	1459.00	2006.00	59.20	1073.36	0.10	24.53	2
1	0	0	1	0	1	8	1459.00	2006.00	59.20	1073.36	0.10	24.53	3
1	0	0	0	0	0	9	1856.00	2320.00	56.40	1067.53	0.10	21.13	1
1	0	0	0	0	1	9	1856.00	2320.00	56.40	1067.53	0.10	21.13	2
1	0	0	0	1	0	9	1856.00	2320.00	56.40	1067.53	0.10	21.13	2
1	0	0	0	1	1	9	1856.00	2320.00	56.40	1067.53	0.10	21.13	2
1	0	0	0	0	0	10	2186.00	2605.00	54.40	1064.19	0.10	19.86	1
1	0	0	0	1	0	10	2186.00	2605.00	54.40	1064.19	0.10	19.86	2
1	0	0	0	1	1	10	2186.00	2605.00	54.40	1064.19	0.10	19.86	2
1	0	0	1	0	1	10	2186.00	2605.00	54.40	1064.19	0.10	19.86	4
1	0	0	1	1	0	10	2186.00	2605.00	54.40	1064.19	0.10	19.86	3
1	0	1	0	1	0	1	31.00	823.00	97.70	1131.44	0.09	63.07	2
1	0	1	0	1	1	1	31.00	823.00	97.70	1131.44	0.09	63.07	2
1	0	1	1	0	0	1	31.00	823.00	97.70	1131.44	0.09	63.07	1
1	0	1	0	0	0	2	98.00	1104.00	89.20	1118.18	0.09	54.40	1
1	0	1	0	1	0	2	98.00	1104.00	89.20	1118.18	0.09	54.40	2
1	0	1	0	1	1	2	98.00	1104.00	89.20	1118.18	0.09	54.40	2
1	0	1	1	0	0	2	98.00	1104.00	89.20	1118.18	0.09	54.40	2
1	0	1	1	1	1	2	98.00	1104.00	89.20	1118.18	0.09	54.40	2
1	0	1	0	0	0	3	190.00	1385.00	84.10	1108.30	0.09	48.76	1
1	0	1	0	0	1	3	190.00	1385.00	84.10	1108.30	0.09	48.76	2
1	0	1	0	0	0	4	303.00	1671.00	79.70	1100.12	0.09	44.02	1
1	0	1	0	1	0	4	303.00	1671.00	79.70	1100.12	0.09	44.02	2
1	0	1	0	1	1	4	303.00	1671.00	79.70	1100.12	0.09	44.02	2
1	0	1	1	1	0	4	303.00	1671.00	79.70	1100.12	0.09	44.02	2
1	0	1	1	1	1	4	303.00	1671.00	79.70	1100.12	0.09	44.02	2
1	0	1	0	0	0	5	472.00	1943.00	75.10	1092.64	0.09	37.72	1
1	0	1	0	1	0	5	472.00	1943.00	75.10	1092.64	0.09	37.72	2
1	0	1	0	1	1	6	720.00	2178.00	71.10	1087.87	0.09	34.08	2
1	0	1	0	1	1	7	1019.00	2440.00	67.70	1083.16	0.09	30.51	2
1	0	1	1	1	0	7	1019.00	2440.00	67.70	1083.16	0.09	30.51	2
1	0	1	0	0	0	8	1400.00	2755.00	64.30	1077.47	0.09	26.64	1
1	0	1	0	0	1	8	1400.00	2755.00	64.30	1077.47	0.09	26.64	2
1	0	1	0	1	0	8	1400.00	2755.00	64.30	1077.47	0.09	26.64	2
1	0	1	0	1	1	8	1400.00	2755.00	64.30	1077.47	0.09	26.64	2
1	0	1	1	1	0	8	1400.00	2755.00	64.30	1077.47	0.09	26.64	2
1	0	1	1	1	1	8	1400.00	2755.00	64.30	1077.47	0.09	26.64	2
1	0	1	0	0	0	9	1812.00	3082.00	61.50	1071.68	0.10	24.10	1
1	0	1	0	0	1	9	1812.00	3082.00	61.50	1071.68	0.10	24.10	2
1	0	1	1	0	0	9	1812.00	3082.00	61.50	1071.68	0.10	24.10	2
1	0	1	1	1	0	9	1812.00	3082.00	61.50	1071.68	0.10	24.10	3
1	0	1	0	1	1	10	2212.00	3394.00	58.90	1067.99	0.10	23.09	3
1	0	1	1	0	0	10	2212.00	3394.00	58.90	1067.99	0.10	23.09	2
1	0	1	1	1	0	10	2212.00	3394.00	58.90	1067.99	0.10	23.09	2
1	1	0	0	0	1	1	23.00	756.00	103.40	1137.14	0.09	66.58	2

1	1	0	0	1	1	1	23.00	756.00	103.40	1137.14	0.09	66.58	2
1	1	0	1	0	0	1	23.00	756.00	103.40	1137.14	0.09	66.58	1
1	1	0	1	1	1	1	23.00	756.00	103.40	1137.14	0.09	66.58	2
1	1	0	0	0	0	2	108.00	921.00	85.90	1115.53	0.09	55.59	6
1	1	0	0	0	1	2	108.00	921.00	85.90	1115.53	0.09	55.59	2
1	1	0	0	1	0	2	108.00	921.00	85.90	1115.53	0.09	55.59	6
1	1	0	1	1	1	2	108.00	921.00	85.90	1115.53	0.09	55.59	2
1	1	0	0	1	0	3	208.00	1078.00	81.70	1106.86	0.09	48.33	8
1	1	0	0	1	1	3	208.00	1078.00	81.70	1106.86	0.09	48.33	2
1	1	0	1	0	0	3	208.00	1078.00	81.70	1106.86	0.09	48.33	4
1	1	0	1	0	1	3	208.00	1078.00	81.70	1106.86	0.09	48.33	3
1	1	0	1	1	1	3	208.00	1078.00	81.70	1106.86	0.09	48.33	2
1	1	0	0	1	1	4	368.00	1285.00	76.60	1096.91	0.09	42.07	2
1	1	0	1	0	0	4	368.00	1285.00	76.60	1096.91	0.09	42.07	3
1	1	0	1	1	1	4	368.00	1285.00	76.60	1096.91	0.09	42.07	2
1	1	0	0	0	0	5	565.00	1547.00	72.90	1091.25	0.09	38.44	1
1	1	0	0	1	0	5	565.00	1547.00	72.90	1091.25	0.09	38.44	2
1	1	0	1	0	0	5	565.00	1547.00	72.90	1091.25	0.09	38.44	4
1	1	0	1	0	1	5	565.00	1547.00	72.90	1091.25	0.09	38.44	3
1	1	0	1	1	1	5	565.00	1547.00	72.90	1091.25	0.09	38.44	3
1	1	0	0	0	1	6	809.00	1828.00	69.60	1086.85	0.09	34.97	2
1	1	0	0	1	1	6	809.00	1828.00	69.60	1086.85	0.09	34.97	2
1	1	0	1	0	1	6	809.00	1828.00	69.60	1086.85	0.09	34.97	3
1	1	0	1	1	1	6	809.00	1828.00	69.60	1086.85	0.09	34.97	6
1	1	0	0	1	1	7	1084.00	2156.00	66.80	1082.46	0.09	30.81	2
1	1	0	1	0	0	7	1084.00	2156.00	66.80	1082.46	0.09	30.81	5
1	1	0	1	0	1	7	1084.00	2156.00	66.80	1082.46	0.09	30.81	5
1	1	0	1	1	1	7	1084.00	2156.00	66.80	1082.46	0.09	30.81	2
1	1	0	0	0	0	8	1393.00	2439.00	63.70	1077.20	0.09	26.94	1
1	1	0	0	1	0	8	1393.00	2439.00	63.70	1077.20	0.09	26.94	2
1	1	0	1	0	1	8	1393.00	2439.00	63.70	1077.20	0.09	26.94	3
1	1	0	0	0	1	9	1813.00	2741.00	60.60	1072.49	0.09	24.93	2
1	1	0	0	1	1	9	1813.00	2741.00	60.60	1072.49	0.09	24.93	2
1	1	0	0	0	0	10	2184.00	3132.00	59.00	1071.27	0.09	22.82	1
1	1	0	0	0	1	10	2184.00	3132.00	59.00	1071.27	0.09	22.82	2
1	1	0	0	1	0	10	2184.00	3132.00	59.00	1071.27	0.09	22.82	2
1	1	0	0	1	1	10	2184.00	3132.00	59.00	1071.27	0.09	22.82	2
1	1	1	0	0	0	1	12.00	832.00	124.80	1159.70	0.09	80.57	1
1	1	1	0	1	1	1	12.00	832.00	124.80	1159.70	0.09	80.57	2
1	1	1	1	0	1	1	12.00	832.00	124.80	1159.70	0.09	80.57	2
1	1	1	0	0	0	2	79.00	1195.00	96.90	1123.74	0.09	58.78	1
1	1	1	0	0	1	2	79.00	1195.00	96.90	1123.74	0.09	58.78	2
1	1	1	0	1	0	2	79.00	1195.00	96.90	1123.74	0.09	58.78	2
1	1	1	1	0	0	2	79.00	1195.00	96.90	1123.74	0.09	58.78	2
1	1	1	1	0	1	2	79.00	1195.00	96.90	1123.74	0.09	58.78	2
1	1	1	1	1	0	2	79.00	1195.00	96.90	1123.74	0.09	58.78	2
1	1	1	0	0	0	3	157.00	1527.00	90.10	1112.11	0.09	52.44	1
1	1	1	0	1	1	3	157.00	1527.00	90.10	1112.11	0.09	52.44	2
1	1	1	1	0	0	3	157.00	1527.00	90.10	1112.11	0.09	52.44	2
1	1	1	1	0	1	3	157.00	1527.00	90.10	1112.11	0.09	52.44	3
1	1	1	0	0	0	4	282.00	1872.00	85.00	1103.41	0.09	47.03	1
1	1	1	0	1	0	4	282.00	1872.00	85.00	1103.41	0.09	47.03	2
1	1	1	1	1	0	4	282.00	1872.00	85.00	1103.41	0.09	47.03	3
1	1	1	0	0	0	5	422.00	2144.00	80.10	1095.61	0.09	41.33	1
1	1	1	0	0	1	5	422.00	2144.00	80.10	1095.61	0.09	41.33	2
1	1	1	1	0	0	5	422.00	2144.00	80.10	1095.61	0.09	41.33	2

1	1	1	1	1	0	5	422.00	2144.00	80.10	1095.61	0.09	41.33	2
1	1	1	1	0	1	6	606.00	2366.00	75.80	1090.12	0.09	37.41	4
1	1	1	0	0	1	7	857.00	2687.00	72.20	1085.87	0.09	31.95	2
1	1	1	0	1	0	7	857.00	2687.00	72.20	1085.87	0.09	31.95	2
1	1	1	1	0	0	7	857.00	2687.00	72.20	1085.87	0.09	31.95	2
1	1	1	1	1	0	7	857.00	2687.00	72.20	1085.87	0.09	31.95	2
1	1	1	0	0	1	8	1216.00	2963.00	68.70	1081.02	0.09	29.00	2
1	1	1	0	1	0	8	1216.00	2963.00	68.70	1081.02	0.09	29.00	2
1	1	1	1	1	0	8	1216.00	2963.00	68.70	1081.02	0.09	29.00	2
1	1	1	0	0	0	9	1684.00	3291.00	65.20	1075.85	0.09	25.96	1
1	1	1	0	0	1	9	1684.00	3291.00	65.20	1075.85	0.09	25.96	2
1	1	1	1	1	0	9	1684.00	3291.00	65.20	1075.85	0.09	25.96	2
1	1	1	0	0	0	10	2116.00	3671.00	62.70	1072.99	0.09	25.08	1
1	1	1	0	1	0	10	2116.00	3671.00	62.70	1072.99	0.09	25.08	2
1	1	1	0	1	1	10	2116.00	3671.00	62.70	1072.99	0.09	25.08	2
1	1	1	1	0	1	10	2116.00	3671.00	62.70	1072.99	0.09	25.08	3
1	1	1	1	1	0	10	2116.00	3671.00	62.70	1072.99	0.09	25.08	2

Appendix D
RE280.NNA - Recall Data for the Neural Network

0 0 0 0 1 1	1	39.00	949.00	18.30	993.85	0.10	13.10	2
0 0 0 1 0 0	1	39.00	949.00	18.30	993.85	0.10	13.10	8
0 0 0 1 0 1	1	39.00	949.00	18.30	993.85	0.10	13.10	5
0 0 0 1 1 0	1	39.00	949.00	18.30	993.85	0.10	13.10	4
0 0 0 1 1 1	1	39.00	949.00	18.30	993.85	0.10	13.10	5
0 0 0 0 1 0	2	238.00	1473.00	18.20	989.23	0.10	10.29	5
0 0 0 0 1 1	2	238.00	1473.00	18.20	989.23	0.10	10.29	2
0 0 0 1 0 1	2	238.00	1473.00	18.20	989.23	0.10	10.29	6
0 0 0 1 1 0	2	238.00	1473.00	18.20	989.23	0.10	10.29	4
0 0 0 1 1 1	2	238.00	1473.00	18.20	989.23	0.10	10.29	5
0 0 0 0 1 0	3	504.00	1956.00	18.30	984.63	0.10	8.24	2
0 0 0 0 1 1	3	504.00	1956.00	18.30	984.63	0.10	8.24	2
0 0 0 1 0 0	3	504.00	1956.00	18.30	984.63	0.10	8.24	5
0 0 0 1 0 1	3	504.00	1956.00	18.30	984.63	0.10	8.24	6
0 0 0 1 1 0	3	504.00	1956.00	18.30	984.63	0.10	8.24	4
0 0 0 1 1 1	3	504.00	1956.00	18.30	984.63	0.10	8.24	5
0 0 0 0 0 1	4	731.00	2525.00	18.40	980.00	0.10	5.66	1
0 0 0 1 1 1	4	731.00	2525.00	18.40	980.00	0.10	5.66	3
0 0 0 0 0 1	5	984.00	3058.00	18.40	980.00	0.10	3.47	1
0 0 0 1 0 1	5	984.00	3058.00	18.40	980.00	0.10	3.47	3
0 0 0 1 1 1	5	984.00	3058.00	18.40	980.00	0.10	3.47	3
0 0 0 0 0 1	6	1264.00	3501.00	18.40	980.00	0.10	1.59	1
0 0 0 0 1 1	6	1264.00	3501.00	18.40	980.00	0.10	1.59	2
0 0 0 1 0 1	6	1264.00	3501.00	18.40	980.00	0.10	1.59	4
0 0 0 1 1 0	6	1264.00	3501.00	18.40	980.00	0.10	1.59	3
0 0 0 1 1 1	6	1264.00	3501.00	18.40	980.00	0.10	1.59	4
0 0 0 0 1 1	7	1550.00	3928.00	18.40	980.00	0.10	0.00	2
0 0 0 1 0 0	7	1550.00	3928.00	18.40	980.00	0.10	0.00	2
0 0 0 1 1 1	7	1550.00	3928.00	18.40	980.00	0.10	0.00	3
0 0 0 0 0 1	8	1966.00	4326.00	18.40	980.00	0.10	0.00	2
0 0 0 1 0 1	8	1966.00	4326.00	18.40	980.00	0.10	0.00	3
0 0 0 1 1 0	8	1966.00	4326.00	18.40	980.00	0.10	0.00	3
0 0 0 0 1 0	9	2367.00	4710.00	18.40	980.00	0.10	0.00	2
0 0 0 1 0 1	9	2367.00	4710.00	18.40	980.00	0.10	0.00	4
0 0 0 1 1 1	9	2367.00	4710.00	18.40	980.00	0.10	0.00	4
0 0 0 0 0 1	10	2754.00	5081.00	18.40	980.00	0.10	0.00	2
0 0 0 1 1 1	10	2754.00	5081.00	18.40	980.00	0.10	0.00	4
0 0 1 0 0 0	1	6.00	804.00	35.10	1014.09	0.10	26.65	1
0 0 1 0 1 0	1	6.00	804.00	35.10	1014.09	0.10	26.65	2

0 0 1 1 0 0	1	6.00	804.00	35.10	1014.09	0.10	26.65	1
0 0 1 1 0 1	1	6.00	804.00	35.10	1014.09	0.10	26.65	2
0 0 1 1 1 1	1	6.00	804.00	35.10	1014.09	0.10	26.65	2
0 0 1 1 1 0	2	70.00	1081.00	25.90	997.48	0.10	15.73	3
0 0 1 0 0 0	3	271.00	1398.00	21.90	989.35	0.10	9.48	1
0 0 1 0 1 1	3	271.00	1398.00	21.90	989.35	0.10	9.48	2
0 0 1 1 0 0	3	271.00	1398.00	21.90	989.35	0.10	9.48	3
0 0 1 0 1 0	4	572.00	1761.00	19.40	982.21	0.10	5.15	2
0 0 1 1 0 1	4	572.00	1761.00	19.40	982.21	0.10	5.15	3
0 0 1 1 1 0	4	572.00	1761.00	19.40	982.21	0.10	5.15	4
0 0 1 1 1 1	4	572.00	1761.00	19.40	982.21	0.10	5.15	4
0 0 1 0 1 0	5	905.00	2181.00	18.40	980.00	0.10	2.78	2
0 0 1 1 0 1	5	905.00	2181.00	18.40	980.00	0.10	2.78	3
0 0 1 1 1 0	5	905.00	2181.00	18.40	980.00	0.10	2.78	3
0 0 1 1 1 1	5	905.00	2181.00	18.40	980.00	0.10	2.78	4
0 0 1 0 1 1	6	1234.00	2654.00	18.40	980.00	0.10	1.35	2
0 0 1 0 1 0	7	1547.00	3108.00	18.40	980.00	0.10	0.00	2
0 0 1 1 1 0	7	1547.00	3108.00	18.40	980.00	0.10	0.00	3
0 0 1 1 1 1	7	1547.00	3108.00	18.40	980.00	0.10	0.00	4
0 0 1 0 1 1	8	1966.00	3541.00	18.40	980.00	0.10	0.00	2
0 0 1 1 0 0	8	1966.00	3541.00	18.40	980.00	0.10	0.00	3
0 0 1 0 0 0	9	2367.00	3956.00	18.40	980.00	0.10	0.00	1
0 0 1 0 1 0	9	2367.00	3956.00	18.40	980.00	0.10	0.00	2
0 0 1 0 1 1	9	2367.00	3956.00	18.40	980.00	0.10	0.00	2
0 0 1 1 0 0	9	2367.00	3956.00	18.40	980.00	0.10	0.00	2
0 0 1 1 0 1	9	2367.00	3956.00	18.40	980.00	0.10	0.00	4
0 0 1 1 1 1	9	2367.00	3956.00	18.40	980.00	0.10	0.00	4
0 0 1 0 1 1	10	2754.00	4351.00	18.40	980.00	0.10	0.00	2
0 0 1 1 0 0	10	2754.00	4351.00	18.40	980.00	0.10	0.00	3
0 0 1 1 1 0	10	2754.00	4351.00	18.40	980.00	0.10	0.00	3
0 1 0 0 0 0	1	0.00	725.00	51.60	1030.50	0.10	38.91	2
0 1 0 0 0 1	1	0.00	725.00	51.60	1030.50	0.10	38.91	2
0 1 0 1 1 1	1	0.00	725.00	51.60	1030.50	0.10	38.91	2
0 1 0 0 0 0	2	167.00	1149.00	28.10	1000.55	0.10	16.97	6
0 1 0 0 0 1	2	167.00	1149.00	28.10	1000.55	0.10	16.97	2
0 1 0 1 1 0	2	167.00	1149.00	28.10	1000.55	0.10	16.97	4
0 1 0 1 1 1	2	167.00	1149.00	28.10	1000.55	0.10	16.97	5
0 1 0 0 0 1	3	428.00	1542.00	25.50	993.56	0.10	13.36	1
0 1 0 0 1 1	3	428.00	1542.00	25.50	993.56	0.10	13.36	2
0 1 0 1 0 1	3	428.00	1542.00	25.50	993.56	0.10	13.36	6
0 1 0 0 0 1	4	648.00	2103.00	23.70	985.85	0.10	9.42	1
0 1 0 1 0 1	4	648.00	2103.00	23.70	985.85	0.10	9.42	3
0 1 0 1 1 0	4	648.00	2103.00	23.70	985.85	0.10	9.42	4
0 1 0 0 0 1	5	884.00	2734.00	23.10	981.31	0.10	6.50	1
0 1 0 1 0 0	5	884.00	2734.00	23.10	981.31	0.10	6.50	3
0 1 0 1 1 1	5	884.00	2734.00	23.10	981.31	0.10	6.50	4
0 1 0 0 0 0	6	1141.00	3329.00	22.50	980.00	0.10	4.16	1
0 1 0 0 1 1	6	1141.00	3329.00	22.50	980.00	0.10	4.16	2
0 1 0 1 0 0	6	1141.00	3329.00	22.50	980.00	0.10	4.16	3
0 1 0 1 0 1	6	1141.00	3329.00	22.50	980.00	0.10	4.16	4
0 1 0 1 1 1	6	1141.00	3329.00	22.50	980.00	0.10	4.16	4
0 1 0 0 0 1	7	1416.00	3861.00	21.90	980.00	0.10	1.89	1
0 1 0 1 0 0	7	1416.00	3861.00	21.90	980.00	0.10	1.89	2
0 1 0 1 0 1	7	1416.00	3861.00	21.90	980.00	0.10	1.89	3
0 1 0 1 1 0	7	1416.00	3861.00	21.90	980.00	0.10	1.89	3
0 1 0 1 1 1	7	1416.00	3861.00	21.90	980.00	0.10	1.89	3

0 1 0 0 0 1	8	1777.00	4349.00	21.80	980.00	0.10	1.09	2
0 1 0 0 1 1	8	1777.00	4349.00	21.80	980.00	0.10	1.09	2
0 1 0 1 1 0	8	1777.00	4349.00	21.80	980.00	0.10	1.09	3
0 1 0 1 1 1	8	1777.00	4349.00	21.80	980.00	0.10	1.09	4
0 1 0 1 0 1	9	2154.00	4797.00	21.80	980.00	0.10	0.69	4
0 1 0 1 1 0	9	2154.00	4797.00	21.80	980.00	0.10	0.69	3
0 1 0 0 0 0	10	2532.00	5229.00	21.80	980.00	0.10	0.44	1
0 1 0 0 1 1	10	2532.00	5229.00	21.80	980.00	0.10	0.44	2
0 1 0 1 0 1	10	2532.00	5229.00	21.80	980.00	0.10	0.44	4
0 1 1 0 0 0	1	0.00	811.00	83.80	1067.16	0.09	62.53	1
0 1 1 1 0 1	1	0.00	811.00	83.80	1067.16	0.09	62.53	1
0 1 1 1 1 1	1	0.00	811.00	83.80	1067.16	0.09	62.53	2
0 1 1 0 0 0	2	40.00	1176.00	37.10	1009.25	0.10	24.68	1
0 1 1 0 0 1	2	40.00	1176.00	37.10	1009.25	0.10	24.68	2
0 1 1 0 1 0	2	40.00	1176.00	37.10	1009.25	0.10	24.68	2
0 1 1 1 0 0	2	40.00	1176.00	37.10	1009.25	0.10	24.68	3
0 1 1 1 1 1	2	40.00	1176.00	37.10	1009.25	0.10	24.68	4
0 1 1 1 0 0	3	182.00	1485.00	30.80	997.69	0.10	16.15	4
0 1 1 1 1 1	3	182.00	1485.00	30.80	997.69	0.10	16.15	5
0 1 1 0 0 0	4	428.00	1876.00	27.30	987.20	0.10	10.77	1
0 1 1 0 0 1	4	428.00	1876.00	27.30	987.20	0.10	10.77	2
0 1 1 1 1 0	4	428.00	1876.00	27.30	987.20	0.10	10.77	4
0 1 1 0 1 0	5	744.00	2260.00	24.10	981.23	0.10	6.35	2
0 1 1 1 0 0	5	744.00	2260.00	24.10	981.23	0.10	6.35	3
0 1 1 1 0 1	5	744.00	2260.00	24.10	981.23	0.10	6.35	3
0 1 1 1 1 1	5	744.00	2260.00	24.10	981.23	0.10	6.35	3
0 1 1 0 0 1	6	1110.00	2687.00	22.10	980.00	0.10	3.62	1
0 1 1 0 1 0	6	1110.00	2687.00	22.10	980.00	0.10	3.62	2
0 1 1 1 0 0	6	1110.00	2687.00	22.10	980.00	0.10	3.62	3
0 1 1 1 0 1	6	1110.00	2687.00	22.10	980.00	0.10	3.62	4
0 1 1 1 1 0	6	1110.00	2687.00	22.10	980.00	0.10	3.62	3
0 1 1 0 0 1	7	1414.00	3220.00	21.80	980.00	0.10	1.81	1
0 1 1 1 1 0	7	1414.00	3220.00	21.80	980.00	0.10	1.81	3
0 1 1 0 0 1	8	1780.00	3742.00	21.70	980.00	0.10	1.08	2
0 1 1 0 1 0	8	1780.00	3742.00	21.70	980.00	0.10	1.08	2
0 1 1 1 0 1	8	1780.00	3742.00	21.70	980.00	0.10	1.08	3
0 1 1 1 1 0	8	1780.00	3742.00	21.70	980.00	0.10	1.08	3
0 1 1 1 1 1	8	1780.00	3742.00	21.70	980.00	0.10	1.08	3
0 1 1 0 0 0	9	2154.00	4217.00	21.80	980.00	0.10	0.69	1
0 1 1 0 0 1	9	2154.00	4217.00	21.80	980.00	0.10	0.69	2
0 1 1 1 0 0	9	2154.00	4217.00	21.80	980.00	0.10	0.69	2
0 1 1 1 0 1	9	2154.00	4217.00	21.80	980.00	0.10	0.69	4
0 1 1 1 1 0	9	2154.00	4217.00	21.80	980.00	0.10	0.69	3
0 1 1 0 0 0	10	2532.00	4672.00	21.80	980.00	0.10	0.44	1
0 1 1 0 1 0	10	2532.00	4672.00	21.80	980.00	0.10	0.44	2
0 1 1 1 0 1	10	2532.00	4672.00	21.80	980.00	0.10	0.44	4
0 1 1 1 1 1	10	2532.00	4672.00	21.80	980.00	0.10	0.44	4
1 0 0 0 1 0	1	44.00	630.00	83.30	1118.48	0.09	55.58	2
1 0 0 0 1 1	1	44.00	630.00	83.30	1118.48	0.09	55.58	2
1 0 0 1 1 0	1	44.00	630.00	83.30	1118.48	0.09	55.58	2
1 0 0 1 1 1	1	44.00	630.00	83.30	1118.48	0.09	55.58	2
1 0 0 0 0 0	2	123.00	748.00	79.20	1109.40	0.09	50.57	6
1 0 0 0 0 1	2	123.00	748.00	79.20	1109.40	0.09	50.57	2
1 0 0 0 1 0	2	123.00	748.00	79.20	1109.40	0.09	50.57	5
1 0 0 0 1 1	2	123.00	748.00	79.20	1109.40	0.09	50.57	2
1 0 0 1 0 1	2	123.00	748.00	79.20	1109.40	0.09	50.57	2

1 0 0 1 1 0	2	123.00	748.00	79.20	1109.40	0.09	50.57	3
1 0 0 0 0 0	3	235.00	914.00	75.50	1102.33	0.09	44.72	7
1 0 0 0 1 0	3	235.00	914.00	75.50	1102.33	0.09	44.72	7
1 0 0 0 1 1	3	235.00	914.00	75.50	1102.33	0.09	44.72	2
1 0 0 1 1 0	3	235.00	914.00	75.50	1102.33	0.09	44.72	4
1 0 0 0 0 1	4	386.00	1084.00	71.80	1095.26	0.09	38.90	2
1 0 0 0 1 0	4	386.00	1084.00	71.80	1095.26	0.09	38.90	7
1 0 0 1 1 1	4	386.00	1084.00	71.80	1095.26	0.09	38.90	3
1 0 0 0 0 1	5	571.00	1270.00	68.80	1090.47	0.09	36.64	2
1 0 0 0 1 0	5	571.00	1270.00	68.80	1090.47	0.09	36.64	2
1 0 0 0 0 0	6	821.00	1502.00	65.20	1085.53	0.09	31.71	1
1 0 0 1 0 1	6	821.00	1502.00	65.20	1085.53	0.09	31.71	4
1 0 0 0 0 0	7	1150.00	1768.00	62.20	1080.29	0.10	28.10	1
1 0 0 0 0 1	7	1150.00	1768.00	62.20	1080.29	0.10	28.10	2
1 0 0 0 1 1	7	1150.00	1768.00	62.20	1080.29	0.10	28.10	2
1 0 0 1 0 1	7	1150.00	1768.00	62.20	1080.29	0.10	28.10	5
1 0 0 1 1 1	7	1150.00	1768.00	62.20	1080.29	0.10	28.10	7
1 0 0 0 0 1	8	1459.00	2006.00	59.20	1073.36	0.10	24.53	2
1 0 0 0 1 0	8	1459.00	2006.00	59.20	1073.36	0.10	24.53	2
1 0 0 1 1 0	8	1459.00	2006.00	59.20	1073.36	0.10	24.53	2
1 0 0 1 1 1	8	1459.00	2006.00	59.20	1073.36	0.10	24.53	4
1 0 0 1 0 1	9	1856.00	2320.00	56.40	1067.53	0.10	21.13	3
1 0 0 1 1 0	9	1856.00	2320.00	56.40	1067.53	0.10	21.13	3
1 0 0 1 1 1	9	1856.00	2320.00	56.40	1067.53	0.10	21.13	3
1 0 0 0 0 1	10	2186.00	2605.00	54.40	1064.19	0.10	19.86	2
1 0 0 1 1 1	10	2186.00	2605.00	54.40	1064.19	0.10	19.86	4
1 0 1 0 0 0	1	31.00	823.00	97.70	1131.44	0.09	63.07	1
1 0 1 0 0 1	1	31.00	823.00	97.70	1131.44	0.09	63.07	3
1 0 1 1 1 0	1	31.00	823.00	97.70	1131.44	0.09	63.07	2
1 0 1 1 1 1	1	31.00	823.00	97.70	1131.44	0.09	63.07	2
1 0 1 0 0 1	2	98.00	1104.00	89.20	1118.18	0.09	54.40	2
1 0 1 1 1 0	2	98.00	1104.00	89.20	1118.18	0.09	54.40	3
1 0 1 0 1 0	3	190.00	1385.00	84.10	1108.30	0.09	48.76	2
1 0 1 0 1 1	3	190.00	1385.00	84.10	1108.30	0.09	48.76	2
1 0 1 1 0 0	3	190.00	1385.00	84.10	1108.30	0.09	48.76	1
1 0 1 1 1 0	3	190.00	1385.00	84.10	1108.30	0.09	48.76	3
1 0 1 1 1 1	3	190.00	1385.00	84.10	1108.30	0.09	48.76	3
1 0 1 0 0 1	4	303.00	1671.00	79.70	1100.12	0.09	44.02	2
1 0 1 1 0 0	4	303.00	1671.00	79.70	1100.12	0.09	44.02	1
1 0 1 0 0 1	5	472.00	1943.00	75.10	1092.64	0.09	37.72	2
1 0 1 0 1 1	5	472.00	1943.00	75.10	1092.64	0.09	37.72	2
1 0 1 1 0 0	5	472.00	1943.00	75.10	1092.64	0.09	37.72	2
1 0 1 1 1 0	5	472.00	1943.00	75.10	1092.64	0.09	37.72	2
1 0 1 1 1 1	5	472.00	1943.00	75.10	1092.64	0.09	37.72	3
1 0 1 0 0 0	6	720.00	2178.00	71.10	1087.87	0.09	34.08	1
1 0 1 0 0 1	6	720.00	2178.00	71.10	1087.87	0.09	34.08	2
1 0 1 0 1 0	6	720.00	2178.00	71.10	1087.87	0.09	34.08	2
1 0 1 1 0 0	6	720.00	2178.00	71.10	1087.87	0.09	34.08	2
1 0 1 1 1 0	6	720.00	2178.00	71.10	1087.87	0.09	34.08	2
1 0 1 1 1 1	6	720.00	2178.00	71.10	1087.87	0.09	34.08	2
1 0 1 0 0 0	7	1019.00	2440.00	67.70	1083.16	0.09	30.51	1
1 0 1 0 0 1	7	1019.00	2440.00	67.70	1083.16	0.09	30.51	2
1 0 1 0 1 0	7	1019.00	2440.00	67.70	1083.16	0.09	30.51	2
1 0 1 1 0 0	7	1019.00	2440.00	67.70	1083.16	0.09	30.51	2
1 0 1 1 1 1	7	1019.00	2440.00	67.70	1083.16	0.09	30.51	3
1 0 1 1 0 0	8	1400.00	2755.00	64.30	1077.47	0.09	26.64	2

1	0	1	0	1	0	9	1812.00	3082.00	61.50	1071.68	0.10	24.10	2
1	0	1	0	1	1	9	1812.00	3082.00	61.50	1071.68	0.10	24.10	2
1	0	1	1	1	1	9	1812.00	3082.00	61.50	1071.68	0.10	24.10	2
1	0	1	0	0	0	10	2212.00	3394.00	58.90	1067.99	0.10	23.09	1
1	0	1	0	0	1	10	2212.00	3394.00	58.90	1067.99	0.10	23.09	2
1	0	1	0	1	0	10	2212.00	3394.00	58.90	1067.99	0.10	23.09	2
1	0	1	1	1	1	10	2212.00	3394.00	58.90	1067.99	0.10	23.09	3
1	1	0	0	0	0	1	23.00	756.00	103.40	1137.14	0.09	66.58	5
1	1	0	0	1	0	1	23.00	756.00	103.40	1137.14	0.09	66.58	5
1	1	0	1	0	1	1	23.00	756.00	103.40	1137.14	0.09	66.58	2
1	1	0	0	1	1	2	108.00	921.00	85.90	1115.53	0.09	55.59	2
1	1	0	1	0	0	2	108.00	921.00	85.90	1115.53	0.09	55.59	2
1	1	0	1	0	1	2	108.00	921.00	85.90	1115.53	0.09	55.59	2
1	1	0	0	0	0	3	208.00	1078.00	81.70	1106.86	0.09	48.33	7
1	1	0	0	0	1	3	208.00	1078.00	81.70	1106.86	0.09	48.33	3
1	1	0	0	0	0	4	368.00	1285.00	76.60	1096.91	0.09	42.07	1
1	1	0	0	0	1	4	368.00	1285.00	76.60	1096.91	0.09	42.07	2
1	1	0	0	1	0	4	368.00	1285.00	76.60	1096.91	0.09	42.07	7
1	1	0	1	0	1	4	368.00	1285.00	76.60	1096.91	0.09	42.07	3
1	1	0	0	0	1	5	565.00	1547.00	72.90	1091.25	0.09	38.44	2
1	1	0	0	1	1	5	565.00	1547.00	72.90	1091.25	0.09	38.44	2
1	1	0	0	0	0	6	809.00	1828.00	69.60	1086.85	0.09	34.97	1
1	1	0	0	1	0	6	809.00	1828.00	69.60	1086.85	0.09	34.97	2
1	1	0	1	0	0	6	809.00	1828.00	69.60	1086.85	0.09	34.97	5
1	1	0	0	0	0	7	1084.00	2156.00	66.80	1082.46	0.09	30.81	1
1	1	0	0	0	1	7	1084.00	2156.00	66.80	1082.46	0.09	30.81	2
1	1	0	0	1	0	7	1084.00	2156.00	66.80	1082.46	0.09	30.81	2
1	1	0	0	0	1	8	1393.00	2439.00	63.70	1077.20	0.09	26.94	2
1	1	0	0	1	1	8	1393.00	2439.00	63.70	1077.20	0.09	26.94	2
1	1	0	1	0	0	8	1393.00	2439.00	63.70	1077.20	0.09	26.94	4
1	1	0	1	1	1	8	1393.00	2439.00	63.70	1077.20	0.09	26.94	4
1	1	0	0	0	0	9	1813.00	2741.00	60.60	1072.49	0.09	24.93	1
1	1	0	0	1	0	9	1813.00	2741.00	60.60	1072.49	0.09	24.93	2
1	1	0	1	0	0	9	1813.00	2741.00	60.60	1072.49	0.09	24.93	3
1	1	0	1	0	1	9	1813.00	2741.00	60.60	1072.49	0.09	24.93	3
1	1	0	1	1	1	9	1813.00	2741.00	60.60	1072.49	0.09	24.93	4
1	1	0	1	0	0	10	2184.00	3132.00	59.00	1071.27	0.09	22.82	2
1	1	0	1	0	1	10	2184.00	3132.00	59.00	1071.27	0.09	22.82	4
1	1	0	1	1	1	10	2184.00	3132.00	59.00	1071.27	0.09	22.82	4
1	1	1	0	0	1	1	12.00	832.00	124.80	1159.70	0.09	80.57	2
1	1	1	0	1	0	1	12.00	832.00	124.80	1159.70	0.09	80.57	2
1	1	1	1	0	0	1	12.00	832.00	124.80	1159.70	0.09	80.57	1
1	1	1	1	1	0	1	12.00	832.00	124.80	1159.70	0.09	80.57	2
1	1	1	0	1	1	2	79.00	1195.00	96.90	1123.74	0.09	58.78	2
1	1	1	0	0	1	3	157.00	1527.00	90.10	1112.11	0.09	52.44	3
1	1	1	0	1	0	3	157.00	1527.00	90.10	1112.11	0.09	52.44	2
1	1	1	1	1	0	3	157.00	1527.00	90.10	1112.11	0.09	52.44	2
1	1	1	0	0	1	4	282.00	1872.00	85.00	1103.41	0.09	47.03	3
1	1	1	0	1	1	4	282.00	1872.00	85.00	1103.41	0.09	47.03	2
1	1	1	1	0	0	4	282.00	1872.00	85.00	1103.41	0.09	47.03	1
1	1	1	1	0	1	4	282.00	1872.00	85.00	1103.41	0.09	47.03	3
1	1	1	0	1	0	5	422.00	2144.00	80.10	1095.61	0.09	41.33	2
1	1	1	0	1	1	5	422.00	2144.00	80.10	1095.61	0.09	41.33	2
1	1	1	1	0	1	5	422.00	2144.00	80.10	1095.61	0.09	41.33	3
1	1	1	0	0	0	6	606.00	2366.00	75.80	1090.12	0.09	37.41	1
1	1	1	0	0	1	6	606.00	2366.00	75.80	1090.12	0.09	37.41	2

1	1	1	0	1	0	6	606.00	2366.00	75.80	1090.12	0.09	37.41	2
1	1	1	0	1	1	6	606.00	2366.00	75.80	1090.12	0.09	37.41	2
1	1	1	1	0	0	6	606.00	2366.00	75.80	1090.12	0.09	37.41	2
1	1	1	1	1	0	6	606.00	2366.00	75.80	1090.12	0.09	37.41	2
1	1	1	0	0	0	7	857.00	2687.00	72.20	1085.87	0.09	31.95	1
1	1	1	0	1	1	7	857.00	2687.00	72.20	1085.87	0.09	31.95	2
1	1	1	1	0	1	7	857.00	2687.00	72.20	1085.87	0.09	31.95	2
1	1	1	0	0	0	8	1216.00	2963.00	68.70	1081.02	0.09	29.00	1
1	1	1	0	1	1	8	1216.00	2963.00	68.70	1081.02	0.09	29.00	2
1	1	1	1	0	0	8	1216.00	2963.00	68.70	1081.02	0.09	29.00	2
1	1	1	1	0	1	8	1216.00	2963.00	68.70	1081.02	0.09	29.00	3
1	1	1	0	1	0	9	1684.00	3291.00	65.20	1075.85	0.09	25.96	2
1	1	1	0	1	1	9	1684.00	3291.00	65.20	1075.85	0.09	25.96	2
1	1	1	1	0	0	9	1684.00	3291.00	65.20	1075.85	0.09	25.96	2
1	1	1	1	0	1	9	1684.00	3291.00	65.20	1075.85	0.09	25.96	3
1	1	1	0	0	1	10	2116.00	3671.00	62.70	1072.99	0.09	25.08	2
1	1	1	1	0	0	10	2116.00	3671.00	62.70	1072.99	0.09	25.08	2

Appendix E
The Original Neural Network Paradigm

Title: Standard Backpropagation, 8 Hidden Nodes, BFS L/R
Display Mode: Network   Type: Hetero-Associative
Display Style: default
Control Strategy: backprop   L/R Schedule: BFS

1 Learn	0 Recall	0 Layer
0 Aux 1	0 Aux 2	0 Aux 3

L/R Schedule: bfs

Recall Step	1	0	0	0	0
Input Clamp	0.00	0.00	0.00	0.00	0.00
Firing Density	100.00	0.00	0.00	0.00	0.00
Temperature	0.00	0.00	0.00	0.00	0.00
Gain	1.00	0.00	0.00	0.00	0.00
Modifier	2.00	0.00	0.00	0.00	0.00
Learn Step	100	200	300	400	0
Coefficient 1	0.50	0.10	0.05	0.01	0.00
Coefficient 2	0.20	0.05	0.01	0.00	0.00
Coefficient 3	0.00	0.00	0.00	0.00	0.00
Temperature	0.00	0.00	0.00	0.00	0.00

IO Parameters
Learn Data: File Rand. (ttrwo13) Binary
Recall Data: File Seq. (trewo13)
Result File: Output
UserIO Program: userio
I/P Ranges: 0.0000, 1.0000
O/P Ranges: 0.0000, 1.0000
I/P Start Col: 1   O/P Start Col: 14
MinMax Table: ttrwo13   # entries: 14

Col:	1	2	3	4	5	6
Min:	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Max:	1	1	1	1	1	1

Col:	7	8	9	10	11	12
Min:	1.0000	0.0000	630.0000	18.2000	980.0000	0.0900
Max:	10	2754	5229	124.8	1159.7	0.1

```
Col:      13       14
Min:  0.0000   1.0000
Max:    80.57        8

Layer: 1
      PEs: 1                              Sum: Sum
  Spcacing: 5     F' offset: 0.00         Transfer: Linear
    Shape: Square                          Output: Direct
    Scale: 1.00    Low Limit: 0.00        Error Func: standard
   Offset: 0.00   High Limit: 9999.00        Learn: --None--
 Init Low: -0.100  Init High: 0.100     L/R Schedule: (Network)
 Winner 1: None                         Winner 2: None
 PE: Bias
    1.000 Error Factor
    0.000 Sum          1.000 Transfer       1.000 Output
        0 Weights      0.000 Error          0.000 Current Error

Layer: In
      PEs: 13                             Sum: Sum
  Spacing: 5     F' offset: 0.00        Transfer: Linear
    Shape: Square                          Output: Direct
    Scale: 1.00    Low Limit: -9999.00     Error Func: standard
   Offset: 0.00   High Limit: 9999.00        Learn: --None--
 Init Low: -0.100  Init High: 0.100     L/R Schedule: (Network)
 Winner 1: None                         Winner 2: None
 PE: 2
    1.000 Error Factor
    0.000 Sum          0.000 Transfer       0.000 Output
        0 Weights      0.000 Error          0.000 Current Error
 PE: 3
    1.000 Error Factor
    0.000 Sum          0.000 Transfer       0.000 Output
        0 Weights      0.000 Error          0.000 Current Error
 PE: 4
    1.000 Error Factor
    0.000 Sum          0.000 Transfer       0.000 Output
        0 Weights      0.000 Error          0.000 Current Error
 PE: 5
    1.000 Error Factor
    0.000 Sum          0.000 Transfer       0.000 Output
        0 Weights      0.000 Error          0.000 Current Error
 PE: 6
    1.000 Error Factor
    0.000 Sum          0.000 Transfer       0.000 Output
        0 Weights      0.000 Error          0.000 Current Error
 PE: 7
    1.000 Error Factor
    0.000 Sum          0.000 Transfer       0.000 Output
        0 Weights      0.000 Error          0.000 Current Error
 PE: 8
    1.000 Error Factor
    0.000 Sum          0.000 Transfer       0.000 Output
        0 Weights      0.000 Error          0.000 Current Error
 PE: 9
    1.000 Error Factor
    0.000 Sum          0.000 Transfer       0.000 Output
```

```
        0 Weights        0.000 Error            0.000 Current Error
  PE: 10
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
        0 Weights        0.000 Error            0.000 Current Error
  PE: 11
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
        0 Weights        0.000 Error            0.000 Current Error
  PE: 12
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
        0 Weights        0.000 Error            0.000 Current Error
  PE: 13
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
        0 Weights        0.000 Error            0.000 Current Error
  PE: 14
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
        0 Weights        0.000 Error            0.000 Current Error

Layer: Hidden 1
      PEs: 8                                  Sum: Sum
   Spacing: 5      F′ offset: 0.00           Transfer: Sigmoid
     Shape: Square                              Output: Direct
     Scale: 1.00     Low Limit: -9999.00      Error Func: standard
    Offset: 0.00   High Limit: 9999.00            Learn: Delta-Rule
  Init Low: -0.100   Init High: 0.100       L/R Schedule: (Network)
  Winner 1: None                              Winner 2: None
  PE: 15
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
       14 Weights         0.000 Error            0.000 Current Error
  PE: 16
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
       14 Weights         0.000 Error            0.000 Current Error
  PE: 17
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
       14 Weights         0.000 Error            0.000 Current Error
  PE: 18
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
       14 Weights         0.000 Error            0.000 Current Error
  PE: 19
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
       14 Weights         0.000 Error            0.000 Current Error
  PE: 20
     1.000 Error Factor
     0.000 Sum            0.000 Transfer          0.000 Output
       14 Weights         0.000 Error            0.000 Current Error
  PE: 21
     1.000 Error Factor
```

```
    0.000 Sum           0.000 Transfer        0.000 Output
      14 Weights        0.000 Error          0.000 Current Error
  PE: 22
    1.000 Error Factor
    0.000 Sum           0.000 Transfer        0.000 Output
      14 Weights        0.000 Error          0.000 Current Error

Layer: Out
     PEs: 1                               Sum: Sum
   Spacing: 5      F' offset: 0.00       Transfer: Sigmoid
    Shape: Square                          Output: Direct
    Scale: 1.00    Low Limit: -9999.00     Error Func: standard
   Offset: 0.00   High Limit: 9999.00         Learn: Delta-Rule
  Init Low: -0.100  Init High: 0.100     L/R Schedule: (Network)
  Winner 1: None                          Winner 2: None
  PE: 23
    1.000 Error Factor
    0.000 Sum           0.000 Transfer        0.000 Output
       9 Weights        0.000 Error          0.000 Current Error
```

Appendix F
BASIC Program EVALUATE.BAS

```
2 REM  PROGRAM EVALUATE.BAS
3 REM
4 REM  THIS BASIC PROGRAM COMPUTES THE AVERAGE
6 REM  DEVIATION FROM 6 OPTIMAL COST AND KANBAN
8 REM  SPECIFICATION FOR A NEURAL NETWORK
9 REM
10 T = 0
20 DIM CST(280, 10), MINCST(280), MAXCST(280), MINKAN(280)
30 CLS
40 COLOR 15, 1
50 INPUT "Enter the file for the output (ext. will be CST) "; OFILE$
60 INPUT "Enter the file for the input "; INFILE$
65 INPUT "Enter the number of recalls "; RECALL
70 IF OFILE$ = "" THEN OFILE$ = "ERASE"
74 JIM$ = OFILE$ + ".PRN"
75 OFILE$ = OFILE$ + ".CST"
80 IF INFILE$ = "" THEN INFILE$ = "NRE280"
95 INFILE$ = "c:\nworks\" + INFILE$ + ".NNR"
110 OPEN "c:\nworks\NRE280.CST" FOR INPUT AS #1
120 OPEN OFILE$ FOR OUTPUT AS #3
130 OPEN INFILE$ FOR INPUT AS #2
140 FOR I = 1 TO 280
150  MINCST(I) = 9999
160  MAXCST(I) = 0
170 FOR KAN = 1 TO 10
180 INPUT #1, CST(I, KAN)
190 IF CST(I,KAN)<MINCST(I) THEN MINCST(I)=CST(I,KAN):MINKAN(I)=KAN
200 IF CST(I, KAN) > MAXCST(I) THEN MAXCST(I) = CST(I, KAN)
210 NEXT KAN
220 NEXT I
225 CLOSE 1
226 OPEN JIM$ FOR OUTPUT AS #1
260 PRINT #3, "Training          COST                     Kanbans"
270 PRINT #3,"Reps  Avg.Dev. Std.  Max    Avg.Dev. Std.  Max "
290 FOR COUNT = 1 TO RECALL
300 FOR Q = 1 TO 3
310 INPUT #2, D$
311 IF D$ = "" THEN INPUT #2, D$
320 NEXT Q
```

```
330 SUM = 0
340 SUMK = 0
350 SUMSQ = 0
355 SUMC = 0
360 SUMSQK = 0
370 MAXCST = 0
380 MAXKAN = 0
390 FOR I = 1 TO 280
400 INPUT #2, M
410 M = INT(M + .5)
420 DIF = (ABS(CST(I, M) - CST(I, MINKAN(I))))
430 IF DIF > MAXCST THEN MAXCST = DIF
440 DIFK = ABS(M - MINKAN(I))
450 IF DIFK > MAXKAN THEN MAXKAN = DIFK
460 SUM = SUM + DIF
470 SUMK = SUMK + DIFK
480 SUMSQ = SUMSQ + (DIF ** 2)
490 SUMSQK = SUMSQK + (DIFK ** 2)
495 LET CD = (DIF / CST(I, MINKAN(I)))
497 LET CK = (DIFK / MINKAN(I))
500 COMPO = 1000 * CD + CK
510 SUMC = SUMC + COMPO
520 NEXT I
530 T = T + 100
550 MEAN = SUM / 280
560 MEANK = SUMK / 280
565 MEANC = SUMC / 280
570 STD = ((SUMSQ - (280 * (MEAN ** 2))) / 280) ** .5
580 STDK = ((SUMSQK - (280 * (MEANK ** 2))) / 280) ** .5
600 PRINT #3, T; MEAN; STD; MAXCST; MEANK; STDK; MAXKAN
605 PRINT #1, USING ".####"; MEAN
610 NEXT COUNT
620 PRINT #3, ""
625 CLOSE 1
628 CLOSE 2
630 CLOSE 3
```

Appendix G
The Refined Neural Network Paradigm

Title: Predictive Back-Prop, 31 Hidden layer, Backprop L/R
Display Mode: Network   Type: Hetero-Associative
Display Style: default
Control Strategy: backprop   L/R Schedule: backprop
1600001 Learn   0 Recall   0 Layer
0 Aux 1   0 Aux 2   0 Aux 3
L/R Schedule: backprop

Recall Step	1	0	0	0	0
Input Clamp	0.0000	0.0000	0.0000	0.0000	0.0000
Firing Density	100.0000	0.0000	0.0000	0.0000	0.0000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000
Gain	1.0000	0.0000	0.0000	0.0000	0.0000
Modifier	1.0000	0.0000	0.0000	0.0000	0.0000
Learn Step	5000	0	0	0	0
Coefficient 1	0.9000	0.0000	0.0000	0.0000	0.0000
Coefficient 2	0.6000	0.0000	0.0000	0.0000	0.0000
Coefficient 3	0.0000	0.0000	0.0000	0.0000	0.0000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000

IO Parameters
Learn Data: File Rand. (ttrwo13) Binary
Recall Data: File Seq. (trewo13)
Result File: Output
UserIO Program: userio
I/P Ranges: 0.0000, 1.0000
O/P Ranges: 0.0000, 1.0000
I/P Start Col: 1   O/P Start Col: 14
MinMax Table: ttrwo13   # entries: 14

Col:	1	2	3	4	5	6
Min:	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Max:	1	1	1	1	1	1

Col:	7	8	9	10	11	12
Min:	1.0000	0.0000	630.0000	18.2000	980.0000	0.0900
Max:	10	2754	5229	124.8	1159.7	0.1

Col:	13	14
Min:	0.0000	1.0000
Max:	80.57	8

```
Layer: 1
     PEs: 1                              Sum: Sum
  Spacing: 5      F' offset: 0.00       Transfer: Linear
    Shape: Square                         Output: Direct
    Scale: 1.00    Low Limit: 0.00      Error Func: standard
   Offset: 0.00   High Limit: 9999.00      Learn: --None--
 Init Low: -0.100  Init High: 0.100   L/R Schedule: (Network)
 Winner 1: None                         Winner 2: None
 PE: Bias
     1.000 Error Factor
     0.000 Sum           1.000 Transfer       1.000 Output
         0 Weights     182.576 Error          0.000 Current Error
Layer: In
     PEs: 13                             Sum: Sum
  Spacing: 5      F' offset: 0.00       Transfer: Linear
    Shape: Square                         Output: Direct
    Scale: 1.00    Low Limit: -9999.00   Error Func: standard
   Offset: 0.00   High Limit: 9999.00      Learn: --None--
 Init Low: -0.100  Init High: 0.100   L/R Schedule: (Network)
 Winner 1: None                         Winner 2: None
 PE: 2
     1.000 Error Factor
     1.000 Sum           1.000 Transfer       1.000 Output
         0 Weights      -0.003 Error          0.000 Current Error
 PE: 3
     1.000 Error Factor
     1.000 Sum           1.000 Transfer       1.000 Output
         0 Weights       0.003 Error          0.000 Current Error
 PE: 4
     1.000 Error Factor
     0.000 Sum           0.000 Transfer       0.000 Output
         0 Weights       0.004 Error          0.000 Current Error
 PE: 5
     1.000 Error Factor
     0.000 Sum           0.000 Transfer       0.000 Output
         0 Weights      -0.003 Error          0.000 Current Error
 PE: 6
     1.000 Error Factor
     1.000 Sum           1.000 Transfer       1.000 Output
         0 Weights      -0.008 Error          0.000 Current Error
 PE: 7
     1.000 Error Factor
     1.000 Sum           1.000 Transfer       1.000 Output
         0 Weights      -0.001 Error          0.000 Current Error
 PE: 8
     1.000 Error Factor
     1.000 Sum           1.000 Transfer       1.000 Output
         0 Weights      -0.011 Error          0.000 Current Error
 PE: 9
     1.000 Error Factor
     0.793 Sum           0.793 Transfer       0.793 Output
         0 Weights       0.001 Error          0.000 Current Error
 PE: 10
     1.000 Error Factor
     0.544 Sum           0.544 Transfer       0.544 Output
         0 Weights       0.008 Error          0.000 Current Error
```

```
PE: 11
   1.000 Error Factor
   0.383 Sum          0.383 Transfer      0.383 Output
       0 Weights      0.007 Error         0.000 Current Error
PE: 12
   1.000 Error Factor
   0.508 Sum          0.508 Transfer      0.508 Output
       0 Weights      0.003 Error         0.000 Current Error
PE: 13
   1.000 Error Factor
   0.000 Sum          0.000 Transfer      0.000 Output
       0 Weights      0.004 Error         0.000 Current Error
PE: 14
   1.000 Error Factor
   0.283 Sum          0.283 Transfer      0.283 Output
       0 Weights      0.009 Error         0.000 Current Error
Layer: Hidden 1
      PEs: 31                                  Sum: Sum
   Spacing: 5      F' offset: 0.00        Transfer: Sigmoid
     Shape: Square                          Output: Direct
     Scale: 1.00   Low Limit: -9999.00  Error Func: standard
    Offset: 0.00  High Limit: 9999.00        Learn: Delta-Rule
  Init Low: -0.100  Init High: 0.100   L/R Schedule: (Network)
  Winner 1: None                          Winner 2: None
PE: 15
   1.000 Error Factor
 -23.577 Sum          0.000 Transfer      0.000 Output
      14 Weights      0.000 Error        -0.002 Current Error
PE: 16
   1.000 Error Factor
 -15.331 Sum          0.000 Transfer      0.000 Output
      14 Weights      0.000 Error        -0.002 Current Error
PE: 17
   1.000 Error Factor
 -19.678 Sum          0.000 Transfer      0.000 Output
      14 Weights      0.000 Error        -0.002 Current Error
PE: 18
   1.000 Error Factor
 -24.278 Sum          0.000 Transfer      0.000 Output
      14 Weights      0.000 Error        -0.002 Current Error
PE: 19
   1.000 Error Factor
   1.810 Sum          0.859 Transfer      0.859 Output
      14 Weights     -0.000 Error        -0.002 Current Error
PE: 20
   1.000 Error Factor
  -9.253 Sum          0.000 Transfer      0.000 Output
      14 Weights     -0.000 Error        -0.002 Current Error
PE: 21
   1.000 Error Factor
 -17.425 Sum          0.000 Transfer      0.000 Output
      14 Weights      0.000 Error        -0.002 Current Error
PE: 22
   1.000 Error Factor
  -7.029 Sum          0.001 Transfer      0.001 Output
      14 Weights     -0.000 Error        -0.002 Current Error
```

PE: 23		
1.000 Error Factor		
-17.763 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 24		
1.000 Error Factor		
-22.701 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 25		
1.000 Error Factor		
-18.570 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 26		
1.000 Error Factor		
-3.438 Sum	0.031 Transfer	0.031 Output
14 Weights	-0.000 Error	-0.002 Current Error
PE: 27		
1.000 Error Factor		
-12.827 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 28		
1.000 Error Factor		
-23.454 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 29		
1.000 Error Factor		
20.634 Sum	1.000 Transfer	1.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 30		
1.000 Error Factor		
-15.464 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 31		
1.000 Error Factor		
-24.666 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 32		
1.000 Error Factor		
-5.332 Sum	0.005 Transfer	0.005 Output
14 Weights	-0.000 Error	-0.002 Current Error
PE: 33		
1.000 Error Factor		
-19.660 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 34		
1.000 Error Factor		
-5.767 Sum	0.003 Transfer	0.003 Output
14 Weights	-0.000 Error	-0.002 Current Error
PE: 35		
1.000 Error Factor		
-31.651 Sum	0.000 Transfer	0.000 Output
14 Weights	0.000 Error	-0.002 Current Error
PE: 36		
1.000 Error Factor		
-3.589 Sum	0.027 Transfer	0.027 Output
14 Weights	-0.000 Error	-0.002 Current Error

```
PE: 37
   1.000 Error Factor
   0.229 Sum           0.557 Transfer      0.557 Output
     14 Weights       -0.001 Error        -0.002 Current Error
PE: 38
   1.000 Error Factor
 -23.643 Sum           0.000 Transfer      0.000 Output
     14 Weights        0.000 Error        -0.002 Current Error
PE: 39
   1.000 Error Factor
 -13.225 Sum           0.000 Transfer      0.000 Output
     14 Weights        0.000 Error        -0.002 Current Error
PE: 40
   1.000 Error Factor
  -9.056 Sum           0.000 Transfer      0.000 Output
     14 Weights       -0.000 Error        -0.002 Current Error
PE: 41
   1.000 Error Factor
  -0.993 Sum           0.270 Transfer      0.270 Output
     14 Weights       -0.000 Error        -0.002 Current Error
PE: 42
   1.000 Error Factor
 -17.897 Sum           0.000 Transfer      0.000 Output
     14 Weights        0.000 Error        -0.002 Current Error
PE: 43
   1.000 Error Factor
  -7.597 Sum           0.001 Transfer      0.001 Output
     14 Weights       -0.000 Error        -0.002 Current Error
PE: 44
   1.000 Error Factor
  -7.854 Sum           0.000 Transfer      0.000 Output
     14 Weights       -0.000 Error        -0.002 Current Error
PE: 45
   1.000 Error Factor
 -26.135 Sum           0.000 Transfer      0.000 Output
     14 Weights        0.000 Error        -0.002 Current Error
Layer: Out
     PEs: 1                              Sum: Sum
  Spacing: 5       F' offset: 0.00      Transfer: Linear
   Shape: Square                         Output: Direct
   Scale: 1.00    Low Limit: -9999.00    Error Func: standard
  Offset: 0.00   High Limit: 9999.00       Learn: --None--
Init Low: -0.100  Init High: 0.100     L/R Schedule: (Network)
Winner 1: None                          Winner 2: None
PE: 46
   1.000 Error Factor
   0.178 Sum           0.178 Transfer      0.178 Output
     31 Weights       -0.035 Error        -0.035 Current Error
```

VITA

The author was born in Roanoke, Virginia on January 17, 1961. He graduated from Franklin County High School in 1979. On May 29, 1983 he received a Bachelor of Science degree in Business Administration from Bridgewater College. In August 1989 he received a Masters Degree in Management Science from Virginia Tech. He also received his Ph.D. in Management Science from Virginia Tech in May, 1992.

In August 1991 he took a position as an Assistant Professor of Production and Decision Sciences at the University of North Carolina at Wilmington.