JOB SHOP SCHEDULING: A PREDICTIVE NEURAL NETWORK MODELING SYSTEM FOR A QUANTIFIED SEQUENCING RULE

By

SINAN SALMAN

Bachelor of Science

Jordan University of Science and Technology

Irbid, Jordan

1999

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in Partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2001

JOB SHOP SCHEDULING: A PREDICTIVE
NEURAL NETWORK MODELING
SYSTEM FOR A QUANTIFIED
SEQUENCING RULE

Thesis Approved:

David B Pratt
Thesis Adviser

Camille DeYong

Manjunath Kamath

Alfred Carlozzi
Dean of the Graduate College

ACKNOWLEDGMENTS

In the name of Allah, the most gracious, the most merciful

I wish to express my sincere indebtedness, and deepest gratitude to my family, especially my parents Mahmoud Salman, Zain Shawwash, and my uncle Suleiman Abu-Suleiman. I thank them for providing me with their unconditional love, guidance and support throughout my life, which made me the person I am today. In particular, I thank my father for his countless sacrifices in which he provided me with a role model I look up to in my life. I also thank my mother and my beloved brothers, Shady, Khalil, and Khalid for their love and friendship.

More than ever, I would like to give my special appreciation to my fiancée, Suzan, for her precious love, tolerance, sacrifice and friendship.

I particularly thank my advisor, Dr. David B. Pratt for his constructive guidance, intelligent supervision, but most of all, for his true friendship in times of frustration and hardship. I also wish to extend my sincere gratitude to my committee members, Dr. Camille DeYong and Dr. Manjunath Kamath whose guidance and suggestions had a significant impact on the quality of this research. I also want thank Dr. Allen C. Schuermann for his help in the simulation part of the research. And I thank Dr. Martin Hagan for his help in the neural networks part of the research.

I would also like to thank the School of Industrial Engineering and Management, and specifically Dr. Camille DeYong for providing me with their generous financial support by giving me the opportunity to work with two of the most professional groups: the Technology Forecasting group and the Advanced Demand Forecasting group.

Finally, I would like to thank my uncle Mohammed Shawwash, and my cousin Amr Abu-Suleiman for their friendship and support during my stay in the states.

TABLE OF CONTENTS

Chapter		Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

BPNN	Backpropagation Neural Network
CR	Critical Ratio
CR_z	Modified Critical Ratio
f	Average Fraction Of Tardy Jobs
FT	Average Flow Time
K	Due Date Allowance Factor
MAD	Mean Absolute Deviation From Due Date
Pa	Processing Time Distribution Average
Pw	Processing Time Distribution Width
RBNN	Radial Basis Neural Network
SI	Average System Inventory
SPT	Shortest Processing Time
TWK	Total Work Content
U	Nominal System Utilization
Z	Modified Critical Ratio Power Factor

I The Problem and its Setting

I.1 Introduction

One of the key factors for a successful organization is the efficient scheduling of activities. An activity is any step required to produce and deliver a product or service. In today's diverse economy, scheduling problems are not limited to manufacturing environments; they can occur in any situation that requires prioritization of activities. It can range from a machining process in a manufacturing environment to processing a customer request in a service environment. In this research, the investigation is limited to manufacturing systems, keeping in mind that the results may also apply to other systems.

One of the most common models for manufacturing systems is the job shop model. Job shop models can be applied to a wide range of manufacturing systems. The assumptions required to apply a job shop model are usually easily justified. A job shop model can be used to describe systems ranging from a machine shop with its steel chips and oil to a hospital with its beds and rooms. Job shops have been an active area of research since people started to study manufacturing as a science. Scheduling activities in job shop systems attract most of the research in this field. Attention to improving job shop scheduling can be justified with the economic benefits realized.

In recent years, Just In Time (JIT) has become a major trend in industry, pushing scheduling research to a new area, Forbidden Early Shipment (FES). Forbidden early shipment requires that if a job finishes early it must wait until its due date before being shipped to the customer. Abu-Suleiman (1997) proposed a modification of a well-known dispatching rule, Critical Ratio (CR), and applied the new rule to a job shop model with forbidden early shipment. The new rule was called the Modified Critical Ratio rule (CR_z). Abu-Suleiman showed that CR_z outperformed other dispatching rules due to its ability to adapt to system conditions using the factor z.

Abu-Suleiman stated in his research that the new proposed rule had a shortcoming; the best value of the factor z was determined empirically. The current research attempts to fill this gap by developing a prediction methodology for the factor z using the function approximation capabilities of neural networks. The proposed methodology will use system configuration parameters as inputs for the neural network and the outcome will be an estimate of the best z factor for the system in question with regard to the performance measure used. Neural networks are known for their superior performance in function approximation compared to regression-based techniques (Haykin, 1999). The purpose of this research is to determine whether neural networks hold promise for addressing the shortcoming identified by Abu-Suleiman in the estimation of z in the CR_z methodology.

This research studies job shops in which the jobs arrive at the shop with pre-assigned due dates in a make-to-order environment. The pre-assigned due date will be used to establish system performance measures. The system will be at the peak of its performance if all the due dates are met without excessive early completions since forbidden early shipment will be included as a system characteristic. The efficient scheduling of the incoming jobs will be the primary concern of this research.

I.2 Finding the "Best" Versus "Optimal" Value of z

In literature, the phrase "optimal value" corresponds to an optimality that can be proven mathematically. In this research, a methodology for predicting an unknown function with unknown characteristics is investigated. This fact prevents a mathematical proof as to the existence of optimality for this function. Thus, this research will only attempt to improve on the current solution in an attempt to push the solution closer to the optimal solution. In doing so the methodology will find the "best value" of z.

I.3 Problem Statement

The selection of a dispatching rule in previous research efforts and in practice is primarily determined by experimentation or relying on previous experience. This research attempts to take a different approach. Using Abu-Suleiman's modified Critical Ratio (CR_z), an attempt will be made to predict the factor z using job shop system configuration parameters. Changing z is equivalent to changing the dispatching rule. Therefore, the objective of this research can be stated as "approximating the best value in the output domain of the Modified CR_z rule using a set of shop configuration parameters for the input range, to facilitate the use of CR_z in manufacturing systems, which is expected to result in better system performance".

II Literature Review

II.1 Introduction

In this chapter, literature in fields related to this research are reviewed. In particular, research in the fields of job scheduling, scheduling in a forbidden early shipment environment, due date assignment, scheduling rules, performance measures, system configuration parameters, and neural networks is examined.

II.2 Literature Review on Types of Job Shops Studied

Scheduling jobs in a job shop environment has been an interest for researchers since the early beginnings of manufacturing science. This can be attributed to the fact that none of the proposed solutions represents an optimal solution under all circumstances.

Eilon and Chowdhury (1976) studied a job shop which consisted of five non-identical machines. Jobs arrived at the job shop in regular batches of 10 jobs. The number of operations required was uniformly distributed between 1 and 5. Operation times were normally distributed with a mean of 20 minutes and a standard deviation of 6 minutes. Routing of a job through the machines was random. A job could have several, but not consecutive, operations on a given machine.

Kanet and Hayya (1982) studied a job shop which consisted of eight unique machines. Random routings were used for the job sequence with each machine being equally likely to be the next machine on a job's routing, repeated operations on a given machine were permitted but no job could have two consecutive operations performed by the same machine. The number of operations was uniformly distributed between 1 and 8. The inter-arrival time of jobs and their operation processing times both followed exponential distributions. The arrival rate was adjusted to produce a 90% nominal utilization level for the job shop.

Rohleder and Scudder (1991) studied a job shop which consisted of nine work centers each with a single machine performing a unique processing function. Random routings were used for the job sequence. The number of operations was uniformly distributed between 2 and 7. The arrival distribution was assumed to be Poisson, with a rate that produced a 90% nominal utilization level for the job shop. Pre-emption, breakdowns, and maintenance times were not considered. Due dates were set using the Total Work Content (TWK) rule with allowances of 3, 6, and 9. Processing times were distributed normally with a mean of 9 hours and a standard deviation of 3 hours. Setup times were included in the processing time with an independent uniform distribution between 0.25 and 2.5 hours.

Philipoom, Rees, and Wiegmann (1994) studied a job shop which consisted of five work centers in sequential order with each work center consisting of two different machines. Processing times were drawn from an exponential distribution with a mean of 1.8 time units. Job inter-arrival times were also selected from an exponential distribution, but with a mean of 1 time unit. This resulted in a shop utilization of 90%.

Abu-Suleiman (1997) studied a job shop which consisted of seven machines. Orders arrived for one unit of each product. Each product was unique. Setup times were included in the processing time, which followed a uniform distribution between 3.5 and 6.5 time units. The number of operations for each job was uniformly distributed between 3 and 7. Routing of jobs was set randomly such that a job had the same chance of visiting any machine except the machine just completed. Inter-arrival times were exponentially distributed. The mean of the exponential distribution was set so that a desired utilization level was achieved.

Sabuncuoglu and Lejmi (1999) studied a job shop that consisted of ten machines. Jobs arrived at the job shop according to a Poisson distribution. The number of operations required for each job was uniformly distributed between 1 and 10. Each operation was equally likely to be performed on any of the ten machines. Processing times were uniformly distributed between 1 and 30 minutes.

Most of the literature in this area studies job shop models with 5 to 10 work centers, with 90% nominal utilization level, and an exponential distribution for the inter-arrival times of jobs. As for the number of operations required for each job to finish, the majority used uniform distributions but they differed on the range, with most using a range similar to [3,7]. Processing times were different from one researcher to another. Also, most literature used random routing, unique machines, and all permitted machine revisiting for a single job, but without visiting the most recently used machine.

II.3 Literature Review on Forbidden Early Shipment

Forbidden Early Shipment (FES) is defined as not allowing an order to be shipped until its pre-established due date has been reached or surpassed (Rohleder and Scudder, 1991). With today's interest in Just In Time (JIT) systems where orders are delivered only when needed, finishing a product early is not desirable. This desire leads to the redefinition of the objectives of the scheduling problem, the new objectives are to minimize both earliness and tardiness in time-based performance measures. Additionally, different objectives are defined for the monetary-based performance measures to incorporate the penalty of holding early finished product.

Rohleder and Scudder (1991) used monetary-based performance measures, specifically Net Present Value (NPV) and Inventory, to schedule jobs in an early forbidden shipment environment. They showed that monetary-based scheduling rules out perform time-based scheduling rules in achieving the economic goals in a manufacturing environment.

Leu (1994) investigated group-scheduling heuristics in a flow shop cellular system with work center sharing for the forbidden early shipment environment. Leu used an extended list of monetary and time based performance measures.

Another approach to modeling the forbidden early shipment problem is to use the due window approach. Sabuncuoglu and Lejmi (1999) used this approach. The research used time-based performance measures. The main idea behind this approach is to redefine the due date point to be a due date interval, this allows for a limited amount of earliness or tardiness without being penalized. In order to facilitate this approach a window function must be defined. A general function is shown in Figure 1.

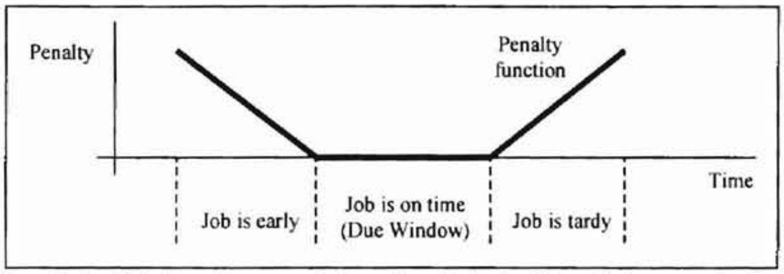

Figure 1 - Due Date Window Approach function *(adapted from Sabuncuoglu & Lejmi, 1999)*

II.4 Literature Review on Due Date Assignment

A standard approach for scheduling jobs in a job shop environment is the use of dispatching rules. Dispatching rules are widely used in scheduling because they are simple and effective heuristics that enable job prioritizing. Research in this area shows that selection of a dispatching rule is highly dependent on the selection of performance measures. Selecting a certain dispatching rule might maximize some set of performance measures, while selecting another dispatching rule might maximize another set of performance measures.

The majority of literature found in this field uses the Total Work Content rule (TWK) to assign due dates to jobs. Eilon and Chowdhury (1976) compared the performance of four due date assignment rules. They studied rules which relied on combinations of the following inputs to generate due dates: job arrival time, expected processing time, expected wait time, and number of jobs waiting in queue to be processed.

Philipoom, Rees, and Wiegmann (1994) proposed a new method that utilizes neural networks to find the best due date assignment. The method computes due dates according to an extended list of inputs which describe the status of the shop. Philipoom, Rees, and Wiegmann also compared the proposed method to conventional regression-based methods found in literature. They concluded that out of three shops studied, neural networks out performed conventional regression-based methods in two shops, and it compared favorably at larger sample sizes in the third.

Sabuncuoglu and Lejmi (1999) stated that there is evidence that the relative performance of priority (scheduling) rules is also affected by the due date tightness, at least for Portion of Tardy jobs (PT) and for Mean Tardiness (MT). This suggests the existence of so called cross over points, with one rule performing best for tighter due dates and another performing best for looser due date. It was found that TWK is the most efficient rule to reduce the cross over effect (Baker, 1984).

In this research we will follow the major trend in research, therefore, the Total Work Content (TWK) due date assignment rule with several tightness levels will be used.

II.5 Literature Review on Scheduling Rules

Scheduling rules, also known as sequencing rules, are heuristics that facilitate the task of scheduling jobs on machines in a manufacturing environment. Researchers have proposed hundreds of scheduling rules that could be used for different situations, and under different conditions and assumptions. A few of the most common, according to Vollmann (1997), are shown in Table 1.

Scheduling Rule	Description
Random	Pick any job in the queue with equal probability
FCFS	First job comes to the queue, is first served
SPT	The Job with Shortest processing time is served first
EDD	The job with the earliest due date is processed first
LWR	The job with the least work remaining is served first
FOR	The job with the fewest operations remaining is served first
CR	Critical Ratio, Priority index = (due date – now) / lead time remaining

Table 1 - Scheduling Rules *(adapted from Vollmann, Berry & Whybark, 1997)*

Rohleder and Scudder (1991) showed in their research that different scheduling rules could be preferred depending on the performance measures used in evaluation. Rohleder and Scudder used Net Present Value (NPV) and inventory to demonstrate the above conclusion. Rohleder and Scudder also examined the relatively poor performance of operation-based due date rules.

Abu-Suleiman (1997) stated that in general, the Critical Ratio rule (CR) has been found to be the rule that performs best in forbidden early shipment in most shop structures. Abu-Suleiman also proposed a modification of the CR rule. The new rule was called the Modified Critical Ratio rule (CR_z). The proposed rule had both continuous range and continuous domain; he used the system configuration as the continuous input range, while the continuous output domain was provided by the output of the CR_z rule as z changed. Abu-Suleiman showed that CR_z out performed other dispatching rules due to its ability to adapt to system conditions using the factor z. This criterion minimized the effect of cross over points found in other dispatching rules. The continuous range was introduced by the z factor, as shown in Equation 1. The z factor can take any real value. Abu-Suleiman showed that the new modification enhanced the scheduling performance for a job shop in a forbidden early shipment environment.

$$CR_z = \frac{DD_j - t}{rp_j^{\,z}}$$

Equation 1

Where:

CR_z : The modified critical ratio

DD_j : The due date for job j

rp_j : The remaining processing time for job j

t : The current time

z : The power factor

According to the CR_z rule, a job with a lower priority index has a higher priority, and should be processed first. The following example demonstrates the effect of using different values for the z factor on the behavior of the selection rule CR_z. Consider the following two values of z: zero and one.

- When z is set to zero, the CR_z rule will behave exactly like the Earliest Due Date (EDD) rule, as shown below:

$$CR_z = \frac{DD_j - t}{rp_j^{\,0}} = DD_j - t$$

Equation 2

- When z is set to one, the CR_z rule will behave exactly like the Critical Ratio (CR) rule, as shown below:

$$CR_z = \frac{DD_j - t}{rp_j^{1}} = \frac{DD_j - t}{rp_j}$$

Equation 3

The above example shows that using appropriate values for the z factor will yield decisions consistent with the EDD and CR rules. Using other values of the z factor yields different decision rules.

Abu-Suleiman used a simulation based search to determine the best z value for a given set of job shop system configuration parameters with regard to a specific performance measure. In this research we will study Abu-Suleiman's modified critical ratio rule, specifically we propose a prediction methodology to predict the z factor based on specific system configurations to improve a specific performance measure. This prediction method should eliminate the need for the simulation based search to determine an appropriate value for the z factor.

II.6 Literature Review on Performance Measures

People use different performance measures for different systems. These measures may have interaction between them, but optimizing one performance measure does not necessarily optimize other performance measures. Because the main goal of this research is to reach the best operating system in order to maximize the economical return, a trade off between performance measures may be necessary to achieve such a system.

Generally, performance measures are classified into two major categories: time based and monetary based. Time based performance measures are used to optimize with regard to throughput, flow time, time in the system, or meeting due dates. Monetary based measures are used to optimize with regard to net present value, total cost per period, system inventory, or relative cost. Monetary performance measures have more obvious economical interpretation, while time based performance measures are easier to translate into physical manufacturing terms.

Philipoom, Rees and Wiegmann (1994) used Mean Absolute Deviation (MAD) and Standard Deviation (SD) of lateness for due dates. Rohleder and Scudder (1991) used Net Present Value (NPV) and inventory performance measures. Kanet and Hayya (1982) used a list of performance measures which included: mean and standard deviation of lateness, fraction of tardy jobs, mean and standard deviation of flow time, maximum job tardiness.

Sabuncuoglu and Lejmi (1999) used Mean Absolute Deviation from due date. They showed that it can be used effectively to optimize job shop systems that apply the due window approach. MAD from due dates out performed Mean Earliness (ME) and Mean Tardiness (MT) for that particular problem.

Abu-Suleiman (1997) used a monetary-based performance measure known as Relative Cost (RC) in addition to other time-based measures, which included: average tardiness and earliness, and average absolute deviation from due date.

Other performance measures found in the literature according to French (1982) included:

– Mean, standard deviation, and max of Make-Span,

– Mean, standard deviation, and max of Tardiness and Earliness,

– Mean, standard deviation, and max of Fraction and Number of Tardy Jobs.

Most of the recent literature in this area studies mean absolute deviation and standard deviation of due date. This is due to the recent growth in interest in Just In Time (JIT) systems. Several researchers used the NPV, which is appropriate when a job's completion time horizon is relatively long.

Monetary based performance measure definitions vary between researchers. This is due to personal preferences in defining costs associated with activities in the job shop. Time based performance measures tend to have agreed definitions among researchers.

II.7 Literature Review on System Parameters

In order to analytically model a job shop system, a set of system configuration parameters must be defined. This allows for distinguishing between different job shop systems using quantitative values associated with system parameters. All system parameters describe manufacturing system conditions.

Most of the literature in the area of job shop scheduling considers only a limited set of system parameters to describe and quantify the change in a proposed system model. The due date tightness factor is one of the most commonly used in literature. System utilization is also widely used in literature.

Lee and Kim (1993) used the following input parameters to quantify the change in their model: due date tightness factor, due date range factor, setup time severity factor, and "number of jobs to number of machines" factor.

Philipoom, Rees and Wiegmann (1994) used an extended list of parameters to describe their proposed model. The list included number of operations required for each job, sum of processing times for each job, sum of jobs currently at the queue on each job's routing, number of operations required to empty the shop of its current workload, and processing time for operations.

Sabuncuoglu and Lejmi (1999) used system utilization and due date tightness in their model. Utilization levels were between 60% and 95%, which is very common in the literature.

In this research a variety of system configuration parameters are utilized. These parameters represent the physical state in the job shop system. These parameters will also be used in the z factor selection for CR_z.

II.8 Literature Review on Neural Networks

The sole inspiration for the invention of *artificial neural networks* was the fact that the human brain operates in an entirely different way from the conventional digital computer (Haykin, 1999). This profound difference is responsible for the human brain's ability to perform certain computations many times faster than the fastest digital computer available today. Haykin defined a neural network as follows:

> *"A neural network is a massively parallel distributed processor made up of simple processing units, which has a neural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:*
>
> 1. *Knowledge is acquired by the network from its environment through a learning process.*
>
> 2. *Inter-neuron connection strengths, known as synaptic weights, are used to store the acquired knowledge." [Haykin, p. 2]*

Caudill (1989) defined a neural network as follows:

> *"A neural network or parallel distributed processing model is a system, consisting of a number of simple, highly interconnected processing elements. The network process information by dynamic response to external inputs." [Caudill]*

In neural networks, knowledge is not stored within individual processing elements, but is represented by the strengths of the connections between elements. Each piece of knowledge is a pattern of activity spread among many processing elements and each processing element can be involved in the partial representation of many pieces of information (Bauer, 1988).

The most common type of Neural Networks according to Hinton (1992) consists of three groups or "layers" of processing elements or "nodes". The first layer is an input layer, which is connected to the hidden layer, which in turn is connected to the output layer. The input layer serves the purpose of feeding raw information to the network, while the output layer produces the outputs. The hidden layer maps input into output using network weights. Weights are the product of the learning process, also known as the training process, in which the neural network is trained using a relatively large sample of data. Figure 2 illustrates the general design of a neural network.

Still, certain precautions should be made, as Billings et al. (1992) points out:

> *"Because the network has been trained by minimizing a cost function... the output of the network will most probably provide a good prediction over the data set used for estimation. Whilst this is almost universally used as a metric of network performance it does not mean that the network is a good model of the underlying system... physically this means that whilst the network will provide good predictions over the data used in training it is valid for that one particular data set and may not provide good predictions for different data sets." [Billings, p. 202-203]*

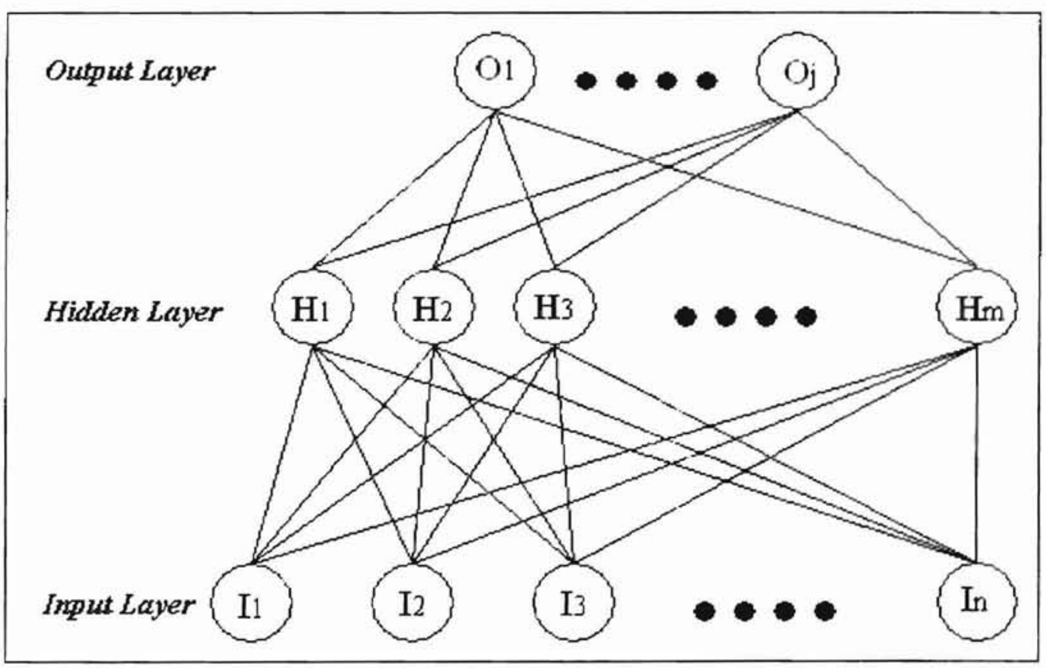

Figure 2 - General Neural Network Design

A relatively new type neural networks is the Radial Basis Network (RBNN). According to Hagan, Demuth, and Beale (1996), it was first introduced in the solution of multi-variable interpolation problems. A radial basis network consists of two layers, the output of first-layer neurons, each representing a basis function, are determined by the distance between the networks input and the "center" of the basis function. As the input moves away from a given center, the neuron output drops off rapidly to zero. The second-layer is linear and produces a weighted sum of the outputs of the first layer.

Another one of the well-known neural networks designs is the Backpropagation Network (BPNN). Backpropagation networks are known for their flexibility, accuracy and wide implementation in different scientific fields. The backpropagation design can use multiple layers with linear and nonlinear transfer functions, which render the design more flexible than the radial basis design.

Neural Networks is a relatively new field, but the application of Neural Networks in the field of scheduling is a trend that is rapidly increasing. Neural Networks are used as a tool for classification and system modeling. Neural networks have been successfully applied in classification problems such as optical recognition, voice recognition, and cancer cells early detection. As for applications in system modeling, neural networks have been successfully applied as an approximation tool where high dimensionality of approximated function prevents the application of classical approximation methods.

Cook and Shannon (1992) applied both regression analysis and neural networks to predict the condition of the end product in a wooden composite board manufacturing process using the manufacturing process parameters as inputs for both techniques.

Regression models could account for only 25% of the variation of these parameters, in contrast to 70% using neural networks. Cook and Shannon stated that the sample first-order autocorrelation coefficients of the inputs were high, pointing to the violation of the uncorrelated regression error assumption. This gave an advantage to neural networks over regression analysis. Cook and Shannon used the SAS regression procedure (PROCREG) and the Backpropagation (BPNN) method, as tools for multi-variable regression analysis and neural network, respectively.

Lee and Kim (1993) proposed a method for scheduling jobs on parallel machines using neural networks. The proposed method makes use of the function approximation properties of backpropagation neural networks to estimate the scaling parameters for a function proposed by Lee and Pinedo (1992). The function calculates a job priority index, which is used to schedule jobs. Input parameters for the neural network included: due date tightness factor, due date range factor, setup time severity factor, and "number of jobs to number of machines" factor.

Philipoom, Rees and Wiegmann (1994) proposed a due date assignment method using neural networks. The new method out performed conventional regression-based due date assignment rules according to the Mean Absolute Deviation (MAD) from due date criterion, but results were mixed for the Standard Deviation of Lateness (SLD) criterion. The conventional rules included Total Work Content (TWK), Number of Operations (NOP), Total Work Content and Number of Operations (TWK+NOP), Jobs in Queue (JIQ), Work in Queue (WIQ), and Response Mapping Rule (RMR). Philipoom, Rees and Wiegmann used SAS in their paper for the regression analysis. Regression analysis was used for comparison purposes.

II.9 Conclusion

In this research, we consider a new job shop scheduling approach for the forbidden early shipment environment. The new approach utilizes the CR_z rule's capability of adapting to different manufacturing situations. This research proposes a new methodology to facilitate the use of CR_z rule in different environments by predicting the z factor based on the system configuration parameters to improve a given performance measure. Literature suggests that neural networks will outperform regression-based prediction techniques. However, multi-variable regression technique will be applied to analyze the data set for comparison purposes.

III Research Goals and Objectives

III.1 Introduction

The modified critical ratio rule CR_z has been shown to perform better than other dispatching rules in most of the cases studied (Abu-Suleiman, 1997). The improvement realized by the use of CR_z rule was up to 5% of the relative cost performance measure. However, a shortcoming of applying the rule was the estimation of the z factor. The approach taken by Abu-Suleiman relied on empirical search methods for each case. This rendered the rule difficult to implement for practical use.

The objective of this research is to find a methodology that will facilitate the use of the Modified Critical Ratio rule (CR_z) in practice. Such a methodology will allow a structured and scientific approach of finding the best z value according to the current job shop conditions. The main goal of this research is to estimate the value of the power factor z as a function of the system configuration parameters, (see Equation 4). The goal is that the estimated value of the z factor will result in the best job shop schedule using the modified critical ratio rule CR_z based on a given performance measure. In addition, more insight into the system configuration parameters interactions and effects on the job shop under question will be realized. Knowing the effects of such interactions on the job shop will help job shop designers focus their improvement efforts where benefits can be maximized.

$$z \cong f(system\_configuration\_parameters)$$

Equation 4

III.2 Tasks

The following tasks are required to accomplish the objective:

- Develop the job shop model.
- Develop, validate, and verify the simulation model.
- Determine using pilot runs the range and the combination of the system configuration parameters to study (i.e. the experimental design).
- Determine the performance measures.
- Perform simulation on the entire range of parameters.
- Analyze the simulation result and determine the best power factor for each combination of system parameters using search methods.
- Develop the neural network model, and perform pilot runs to determine the best neural network configuration for estimating the z factor.
- Train the neural network design of choice for estimating z as a function of system parameters.
- Develop a multiple regression model for estimating z for comparison (benchmarking) purposes.
- Verify the z estimation method by comparing estimates to actual values.
- Compare results of neural networks with results of multiple regression.
- Develop conclusions and recommendations.
- Document the research.
- Propose future areas of research.

IV Research Methodology

IV.1 Introduction

This research uses a variety of tools to accomplish the desired objectives. In this chapter specific details of the proposed methodology are discussed. Research fields that benefited this research include: simulation, neural networks, and multiple regression.

IV.2 Methodology Overview

This section outlines the proposed methodology. The first step in building the proposed methodology is to define the job shop model to be studied. The second step is to build a simulation model of the job shop model. The model will include system configuration parameters as input to the simulation and performance measures as the outputs from the simulation. The next step is to validate and verify the simulation model. Next, the defined set of system parameters combinations is passed into the simulation model to generate the required output performance measures. The generated output is then analyzed, and a z factor is assigned for each of the job shop cases studied using the search method described in Abu-Suleiman's thesis.

When the input dataset is finalized, the approximation part of the methodology is conducted. A neural network is designed, trained, and tested. The next step is to apply the neural network and obtain estimates of z. The estimations are then compared to values obtained using search methods. The final step is to develop the multiple regression model, and compare it to neural networks results, in addition to the search results. The quality of the estimates will be judged by the percentage of cases justified by the approximation method, as Cook and Shannon (1992) used in their research.

After comparing results, drawing conclusions, and making adjustments, the methodology will be ready to apply in day-to-day real world application of the studied job shop model. The following sections describe in more detail the different parts of the proposed methodology.

IV.3 Job Shop Description

The job shop model used in this research was developed to be consistent with similar models found in the literature, especially Abu-Suleiman's thesis. The model consists of seven unique machines. Jobs arrive at the job shop in batches of one job per batch. The inter-arrival time for jobs is exponentially distributed. The mean of the inter-arrival time is set so that a predetermined nominal utilization level is achieved. Each job is unique, and therefore requires a setup time, which is included in the processing time. The number of operations required for each job to complete its route is sampled from a discrete uniform distribution between 3 and 7 operations. Job routing is random; each machine is equally likely to be the next machine on a job's routing. A job can visit the same machine more than once on its route, but no job can have two consecutive operations performed by the same machine. Processing time is sampled from several uniform distributions in which means vary between 5 and 20 time units, and interval widths vary between 10% and 40% of the distribution mean based on the experimental design. The processing time distribution is varied to allow the modeling of various variability levels in the system. All jobs are scheduled using the modified critical ratio rule CR_z. Nominal utilization levels are achieved by setting the mean of the inter-arrival time for jobs using Equation 5 (Abu-Suleiman, 1997). Equation 5 takes into consideration the special routing condition stated earlier. The condition prohibits jobs from revisiting the same machine consecutively, hence the factor 1.4. A complete derivation for Equation 5 can be found in Abu-Suleiman's thesis.

$$\lambda_o = \frac{1.4\rho}{\mu}$$

Equation 5

Where:

λ_o: The order inter-arrival time

ρ: Desired nominal system utilization

μ: Average processing time

Job due date assignment is accomplished using the Total Work Content method (TWK). The total work content method is common in related literature. This research will use a wide range of tightness levels in addition to other system parameters to describe the different job shops. The Total Work Content rule (TWK) is defined as follows:

$$DD_i = a_i + k\sum_{i=1}^{n} P_{i,j}$$

Equation 6

Where:

DD_j : Due date of job j

a_j : Arrival time of job j

$P_{i,j}$: Processing time of operation i for job j

k : Allowance factor

n : Number of operations

IV.3.1 Assumptions

The following assumption are made throughout this research:

- Each job is an entity; no two operations on the same job may be processed simultaneously,
- No machine can process more than one job at a time,
- No blocking; in process inventory is allowed,
- No machine breakdowns, preemption, scrap, rework, or job cancellation,
- Time to move between jobs is negligible,
- There is no delay between the receipt of product orders and job releases,
- Due date is set at job release,
- Time Value of Money (TVM) is not considered.

IV.3.2 System Parameters

The set of system configuration parameters is one of the key factors for this research. It is common knowledge that the best approximating method can only perform as well as the quality of the provided input. Therefore, selecting the correct combination of system configuration parameters becomes a critical step in this research. The selected system configuration parameters will be the inputs for the approximation stage in this research. The approximation stage includes neural network and multiple regression analysis. The following is a list of the system parameters to be considered:

- System utilization (u): system utilization will be used to determine the arrival rate of jobs into the system, based on Equation 5.
- Due date allowance factor (k): k is the factor used in Equation 6 to determine the tightness level used for determining the due date assignment.
- Processing time distribution average (Pa): Pa will be used in Equation 5 to determine the arrival rate of jobs into the system (μ in Equation 5). It will also describe the flexibility of the system in terms of scheduling jobs; large values of Pa will result in smaller number of jobs and longer processing times, which will result in less flexibility in scheduling jobs.
- Processing time distribution width (Pw): Pw will determine the variability of the system. A large Pw value will result in large differences between processing times of different jobs, the result will be a highly variable system with larger number of jobs waiting in queues and longer flow times.

IV.3.3 Performance Measures

This research will use a list of benchmarks that will consist of both inventory-based and time-based performance measures. The reason for including both types of measures is to generalize the proposed methodology as much as possible. The performance measures used in this research will include the following:

- Average flow time (*FT*),
- Average fraction of tardy jobs (*f*),
- Average system inventory (*SI*), and
- Mean absolute deviation from due date (*MAD*).

IV.4 The Simulation Model

ARENA will be the simulation tool of choice for this research. ARENA is a graphical user interface for the popular simulation language, SIMAN. In order to build the simulation model, the job shop model that will be used in this research needs to be developed. This was described in the previous section. Simulation will be used to provide the necessary input for the neural network training process which is a part of the proposed methodology. Specifically, simulation will generate the system performance measures associated with a specific set of system configuration parameters. By varying the value of z for each combination of system parameters, the best corresponding z value according to a given performance measure will be found with the help of a search method. Appendix A contains a detailed discussion of the simulation model.

After building the simulation model, validation and verification is performed to insure accurate representation of the actual job shop model. The validation process of the simulation model includes a comparison with Abu-Suleiman's results as well as performing load analysis on the system. As for the comparison of result, three performance measures were included due to their availability in Abu-Suleiman's research, the measures are: average earliness, average tardiness, and mean absolute deviation from due date.

After a comprehensive study it was found that some discrepancies exist in trends when comparing average tardiness and mean absolute deviation of due date with Abu-Suleiman's results. However, ranges of values are comparable in both cases, and a very close match was noticed when comparing average earliness. The discrepancies can be attributed to the fact that Abu-Suleiman used a different method to measure average tardiness and average earliness. Abu-Suleiman used the total number of jobs finished at the end of the simulation to calculate averages, where in this research only the total number of tardy jobs is used to calculate average conditional tardiness (only jobs that are tardy are included in the average), as well as using total number of early jobs to calculate average conditional earliness. Due to the unavailability of the simulation code in Abu-Suleiman's research, further investigation in this issue was not an option. Instead additional verification and load analysis was done to insure a representative simulation model. The load analysis verified that the simulation model outputs and statistics were accurate and correct.

The verification process included a full trace of several entities (jobs). The trace of variables and attributes was checked against hand-calculated results to insure correctness of the model. The route of each job and the behavior of each station were observed to insure accurate depiction of the job shop model. Appendix B contains an example trace record for a tardy job.

There are three important characteristics for a simulation model: the warm-up period, the run length, and the number of replications. The latter two characteristics determine the simulation statistics precision. Increasing the number of replications will increase the statistical sample size, while increasing the run length decreases the variability of each within-run average. In both cases more precision is achieved, therefore, only the run length was investigated, and number of replications was fixed at five replications per case.

In this research flow time is used to determine the simulation model characteristics. To determine the warm-up period, the procedure discussed in Law and Kelton (1991) was used. Both Abu-Suleiman (1997) and Widjaja (1997) used this method in their research. Figure 3 shows the moving average of flow time in the most extreme job shop case:

- Processing time distribution average, Pa = 20 time units
- Processing time distribution interval width, Pw = 8 time units
- Due date allowance factor, K = 3
- Nominal system utilization, U = 90%

The listed parameters insure that the worst case in the range of system configuration parameters is used for warm-up analysis. The list describes a high utilization job shop with high variability, and tight due date allowance factor. The Critical Ratio (CR) scheduling rule was used for both warm-up and run length analyses.

Figure 4 shows an enlarged version of Figure 3 for the first 4000 jobs. The graph illustrates how the flow time stabilizes after the first 700 jobs (approximately) are finished. Therefore, the warm-up period was set to the first 900 jobs to include a safety factor. This corresponds to approximately 15,000 time units.

As for the run length analysis, it was achieved by plotting weighted errors of average flow time versus the number of jobs simulated. The weighted average was achieved by dividing the 95% half-width confidence interval over the average flow time as illustrated in Figure 5. The number of finished jobs determined the simulation run length. This approach eliminated the effect of the job processing time on simulation time.

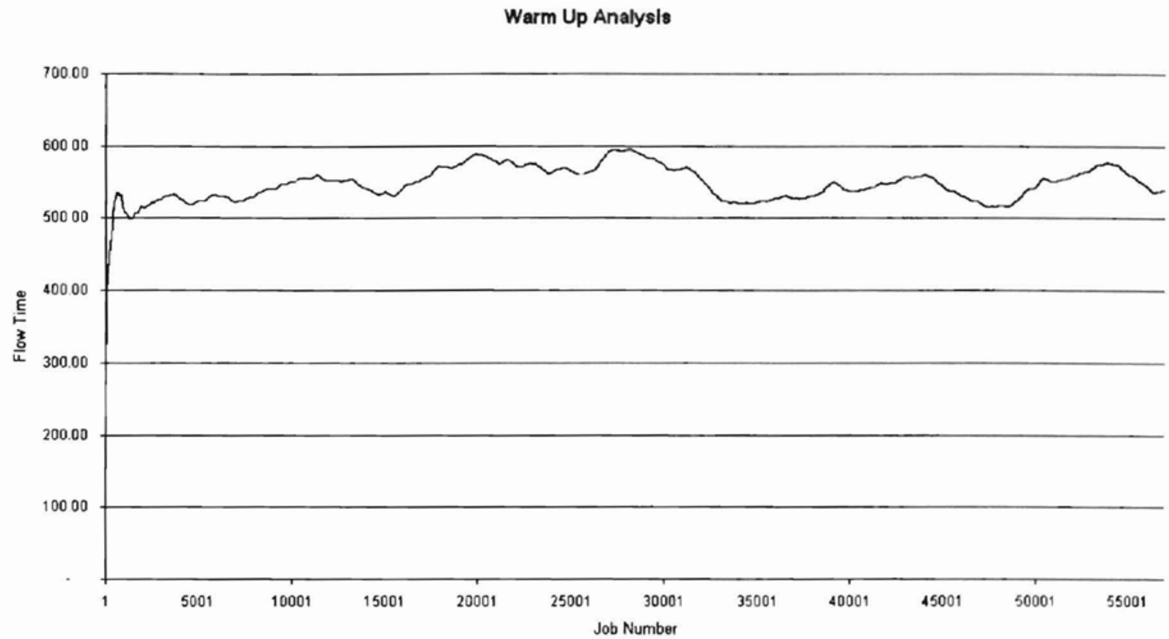

Figure 3 - The moving average of flow time in the most extreme job shop case

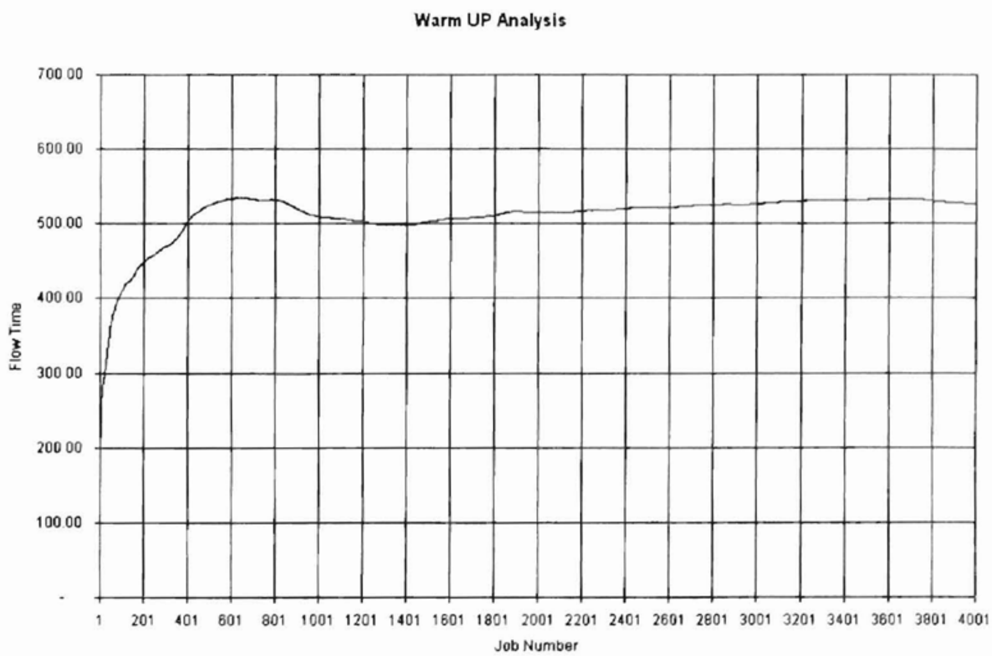

Figure 4 - An enlarged version of Figure 3 for the first 4000 jobs

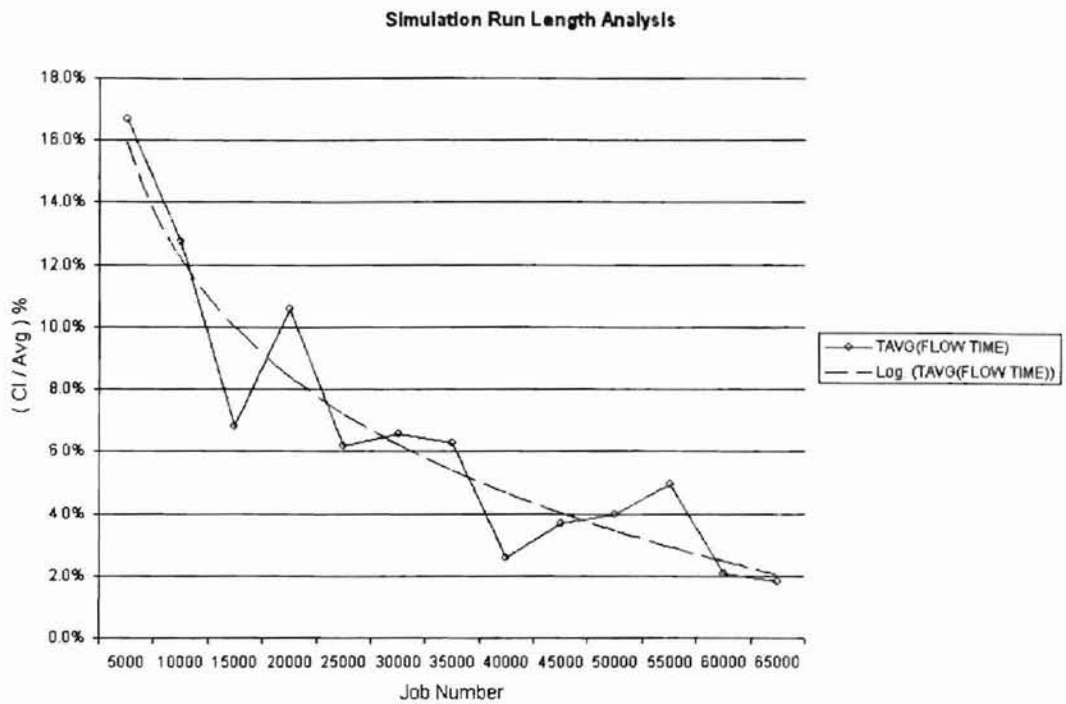

Figure 5 - Weighted error of average flow time Vs. run length

Figure 5 shows that there is a trade-off between the error level allowed and simulation time. Because the simulation time in this research is critical due to the large number of cases studied, an error level of 4% will be allowed. This will correspond to simulating 45,000 jobs. The solid line represents the average flow time statistic using the Arena tally average function TAVG.

One of the challenging factors in this research was the total simulation time. Due to the large number of cases simulated, any small increase in the individual case simulation time meant a very large increase in the total simulation time. Table 2 shows the ranges of each system configuration parameter used in this research. By forming all possible combinations from the listed values, Table 3 shows the total number of cases simulated and the time needed for the total simulation to finish. It was found that on an Intel Pentium III PC, it required more than 10 days to finish the simulation runs. The full list of cases were broken into 5 groups, and simulated on 5 machines, this procedure reduced the total simulation time to a little over 2 days.

	MIN	MAX	Step	Number of cases
Processing Time Average (Time Units)	5	20	7.5	3
Processing time width (% of Avg.)	10%	40%	15%	3
Due Date Allowance Factor k	3	9	1	7
Nominal System Utilization u	60%	90%	2.5%	13
Power Factor z	-1	2.5	0.25	15

Table 2 - Ranges of system configuration parameters used

Total number of unique cases	12285
Number of replications per case	5
Total number of simulations	61425
Total number of unique job shops	819
Simulation time per replication (min)	0.24
Total time required (Days)	10.17
Number of parallel PCs	5
Total time required per PC (Days)	2.03

Table 3 - Total number of cases simulated and simulation time requirements

After obtaining the simulation results, the results were analyzed and the best z values were selected using an exhaustive search method. The search produced the best values of z for each performance measure. A sample of the results can be found in Appendix E, Table 10.

The next step is the prediction process, in which multiple regression and neural networks are evaluated.

IV.5 Regression Analysis

Multiple regression analysis was applied using SAS software (SAS, 1990). SAS/STAT is the research standard software in the field of statistical analysis. Multiple regression analysis using a polynomial of the fourth degree was used to provide a benchmark for the overall performance of the neural network analysis proposed in this research. Using a fourth order polynomial to regress four different variables (K, U, Pa, Pw) provided 69 degrees of freedom. For the regression process, only 40% of the data was used, and the rest of the data (60%) was left to test the quality of the regression model. This division was used to ensure generalization of the regression model over non-regressed data points. The SAS code and the resulting regression model can be found in Appendix C.

IV.6 Neural Network

In this research two neural network designs were investigated. After comparing the performance of both designs, only one neural network design was selected, trained, tested, and applied to the simulation results. The next step was to verify the estimating methodology by comparing predicted z values to actual z values for all system parameter combinations on the four performance measures studied.

Neural networks were used to model the behavior of z as a function of system parameters. Specifically, Radial-Basis Neural Networks (RBNN) and Backpropagation Neural Networks (BPNN) were used to approximate the function f (see Equation 4). Neural networks have several advantages over other function approximation methods. One advantage is the superiority of neural networks when dealing with higher dimension problems, like the one studied in this research. When several system parameters are included, many approximation methods work poorly or not at all for such problems. RBNNs are known to perform better than other neural networks and numerical methods when used in approximating functions because they are immune to the curse of dimensionality; the rate of convergence is independent of the input dimensionality (Haykin, p. 291-292). Still, RBNN are difficult to generalize over the entire input range; RBNNs tend to "over-fit" the approximated function resulting in an almost perfect approximation over the training range, but with poor generalization over the testing range. On the other hand, BPNNs are a more flexible design. Using the generalization techniques described in the "Neural Networks Toolbox for use with MatLab" reference results in a good generalization over the entire data set. The disadvantage to using the BPNN design was that training the network took more time than RBNN. Furthermore, there are more design parameters to be considered than RBNN.

MatLab is the tool for applying neural network analysis. MatLab is a high-performance language for technical computing. It also includes many Toolboxes, which eliminate most of the overhead tied to programming. MatLab also includes a very flexible neural network toolbox that will facilitate the use of neural networks in this research. Appendix D includes MatLab code, results, and a technique discussion.

Because BPNN proved to outperform RBNN in the pilot runs, code and discussion of code is limited to BPNN. A comparison between RBNN and BPNN results is presented in Chapter V.

V Results

In this chapter results of the regression and neural network approximation methods are presented and discussed.

V.1 Multiple Polynomial Regression Approximation

The regression model used in this research included a polynomial function of the fourth order with four independent variables (U, K, Pa, Pw). This model insures that full interaction between the four different system parameters is taken into consideration. Table 4 shows a summary of the results of the regression analysis.

The simulation data was divided into two sets: a training set and a testing set, which consisted of 40% and 60% of the entire data, respectively. This division allows the measurement of generalization of the regression model; that is how well the model predicts z values over non-regressed data points. It is obvious that the regression model is not adequate for either *SI* or *FT* based on the low values for r^2. Furthermore, r^2 values for *MAD* and *f* are not acceptable for practical use; the model explained only 65.0% and 53.7% of the predictions for *MAD* and *f*, respectively.

	r^2 for training set (40%)	r^2 for testing set (60%)
MAD from Due Date (*MAD*)	0.779	0.650
System Inventory (*SI*)	0.316	0.136
Flow Time (*FT*)	0.388	0.045
Fraction of Tardy Jobs (*f*)	0.733	0.537

Table 4 - Multiple polynomial regression analysis results

V.2 Neural Networks Approximation

Two neural network designs were used in this research, RBNN and BPNN. For the RBNN design, there are two parameters that determined the architecture: the spread of neurons and the maximum number of neurons allowed. The spread parameter determines the bias radius of the radial basis function, this allows the adjustment of the function sensitivity to inputs. A higher spread value means that the radial basis function will respond to a larger range of inputs. The maximum number of neurons parameter determines the size of the network. A large network will perform better on the training set but might generalize poorly on the testing set, and a trade-off is necessary to obtain a well-generalized solution.

Although there are general guidelines for designing neural networks, there are no specific rules of thumb for neural networks design. Neural network design is an iterative process of trial and error. in which experience plays an important rule. Table 5 shows the results of applying RBNN for the prediction of z values for different performance measures.

	r^2 for training set (40%)	r^2 for testing set (60%)	Spread	Max Neurons
MAD from Due Date (*MAD*)	0.796	0.759	10	30
System Inventory (*SI*)	0.269	0.197	1.1	16
Flow Time (*FT*)	0.286	0.140	6	19
Fraction of Tardy Jobs (*f*)	0.636	0.640	25	2

Table 5 - RBNN results and architecture parameters

It is obvious that RBNN outperformed regression prediction by comparing the r^2 values in Table 4 and Table 5. This is especially true in the case of *MAD* and *f*. As for *SI* and *f* there was a slight improvement, but still not enough to justify the use of RBNN.

The next step was to apply the BPNN design. BPNNs usually have more flexibility than RBNNs, but at the added cost of more architecture parameters to be determined. BPNNs can be designed with different numbers of hidden layers, in which the number of neurons can also vary. There are several algorithms available to train BPNNs. Finally, BPNNs can be designed with different transfer functions for each hidden layer, which adds more flexibility to the neural network architecture. Refer to Appendix D for more information about the BPNN design and architectures used in this research.

All the BPNN architectures used in this research consisted of two hidden layers, in which the "tansig" transfer function was used. As for the output layer a "purelin" transfer function was used (Math Works, 2000). Table 6 shows the BPNN results and hidden layers architecture parameter.

BPNNs proved to outperform both the RBNNs and multiple regression methods. BPNNs could predict *MAD* with a high accuracy of 86.5%. As for *FT* and *f*, BPNN predicted z values with a conservative accuracy. Still BPNNs predicted *FT* with an accuracy of 52.6% in contrast with 14.0% using RBNNs. In the case of *SI*, the accuracy was too low to be accepted, although it was higher than regression and RBNN.

	r^2 for entire data	Hidden layers architecture
MAD from Due Date (*MAD*)	0.865	4-3-1
System Inventory (*SI*)	0.208	3-3-1
Flow Time (*FT*)	0.526	6-3-1
Fraction of Tardy Jobs (*f*)	0.655	3-2-1

Table 6 - BPNN results and hidden layers architecture parameter

Further analysis of the results was only conducted on BPNN results because of its superior performance over regression and RBNN methods. BPNN results were investigated to discover the sensitivity of each performance measure to prediction error in z estimations. Performance measures were obtained by linear interpolation performed on simulation data. The interpolated performance measures corresponding to the predicted z values were then compared to actual performance measures, which corresponded to the best z values. The comparison process was done by calculating and normalizing absolute errors in z values and performance measures. The normalization was done to provide an unbiased picture of the error in estimates. Table 7 shows the error statistics for predicting z values. Table 7 shows that BPNN predicted z value for *f* with small error average and small standard deviation. As for predicting z values for *MAD*, the error average and standard deviation came second to those of *f*. Error in predicting z values from *FT* and *SI* followed. Minimum and maximum errors provide a worse case/best case measure for the prediction methodology. The normalized error in z prediction was defined as the absolute difference between the best z value and the predicted value, divided by the studied z range, which was equal to 3.5 in this research.

	Statistics of absolute normalized error in prediction of z values			
	MAD	*SI*	*FT*	*f*
Minimum error	0.0%	0.0%	0.0%	0.0%
Maximum error	32.5%	80.6%	71.4%	25.2%
Average error	8.0%	12.3%	9.4%	5.0%
Standard deviation of error	6.7%	10.5%	8.0%	4.1%

Table 7 - Error statistics for predicting z values

It was found that having a low error in predicting z values does not necessarily guarantee low error in achieved performance measures. This is due to the fact that different performance measures have different sensitivity levels to errors in z value predictions. This fact required further investigation of errors found in performance measures corresponding to predicted z values. Table 8 shows the error statistics in performance measures due to errors in predicting z values. Although predicted z values for *MAD* came second to those for *f* judging by the error statistics, the corresponding error in *MAD* was significantly lower. This is due to the relatively low sensitivity of *MAD* to errors in predicted z values, and the relatively high sensitivity of *f* to errors in predicted z values. Table 8 also shows that errors in predicting z values have a cascading effect on both *SI* and *FT*. In the case of *f*, it was found that most of the predictions had low normalized absolute error, still there were few cases with large normalized absolute error, hence the low average with high standard deviation and maximum statistics.

	Statistics of absolute normalized error in performance measures due to error in prediction of z values			
	MAD	*SI*	*FT*	*f*
Minimum error	0.0%	0.1%	0.1%	0.0%
Maximum error	7.4%	99.8%	99.9%	54.8%
Average error	0.4%	51.3%	48.3%	3.7%
Standard deviation of error	0.7%	27.9%	33.4%	8.3%

Table 8 - Error statistics in performance measures due to error in predicting z values

The next step in results analysis was to determine the feasibility of applying the neural network approximation method. In other words, what is the performance improvement realized by applying CR_z coupled with the neural network z approximation method?

In order to answer this question, a comparison based on the same job shop model but with different sequencing rules was carried out. CR_z results were compared to CR and

SPT results using the same parameters and the same simulation model. This comparison was restricted to extreme cases; only the maximum and the minimum of each system configuration parameter were considered. There are 16 different combinations resulting from the described parameters list.

Appendix E contains a full listing of the parameters of the compared cases and the corresponding performance measures using the CR_z, CR, and SPT sequencing rules. In addition, Appendix E contains a comparison between CR_z, CR and SPT. Table 9 summarizes the results of this comparison.

	Statistics of Normalized difference in performance measures due to change of sequencing rule to CR. $(Pcr-Pcr_z)/Pcr_z$				Statistics of Normalized difference in performance measures due to change of sequencing rule to SPT. $(Pspt-Pcr_z)/Pcr_z$			
	MAD	*SI*	*FT*	*f*	*MAD*	*SI*	*FT*	*f*
Minimum error	0.2%	0.2%	0.1%	0.0%	1.5%	0.2%	0.1%	-17.4%
Maximum error	33.6%	16.1%	15.8%	9.7%	24.2%	7.5%	7.3%	2.8%
Average error	10.8%	6.8%	6.6%	3.8%	8.3%	2.4%	2.2%	-3.4%
Standard deviation of error	11.6%	6.2%	6.2%	3.6%	7.5%	2.4%	2.4%	8.1%

Table 9 - Comparison results between CRz, CR, and SPT sequencing rules

It is obvious from Table 9 that CR_z performance was superior to both CR and SPT in all performance measures studied except *f*. SPT performed slightly better than CR_z in this case. This can be attributed to the fact that extremely high system utilization produced relatively large values of *f*, which have a biased effect on the average and standard deviation due to their large magnitude although they are only a few cases.

Finally, best case/worse case analysis was conducted. This analysis approach attempts to give a general overview of the previously presented results. Best case/worst case values correspond to minimum and maximum values found in the previously mentioned 16 cases. Error range was calculated using 3 standard deviation intervals; this results in a range that includes 99.7% of the data based on a normal distribution. Figure 6 through Figure 9 illustrate these results by fixing performance measures obtained using CR or SPT sequencing rules at 100%, and then adding improvement introduced by the use of the CR_z rule, then adding 3-sigma error intervals.

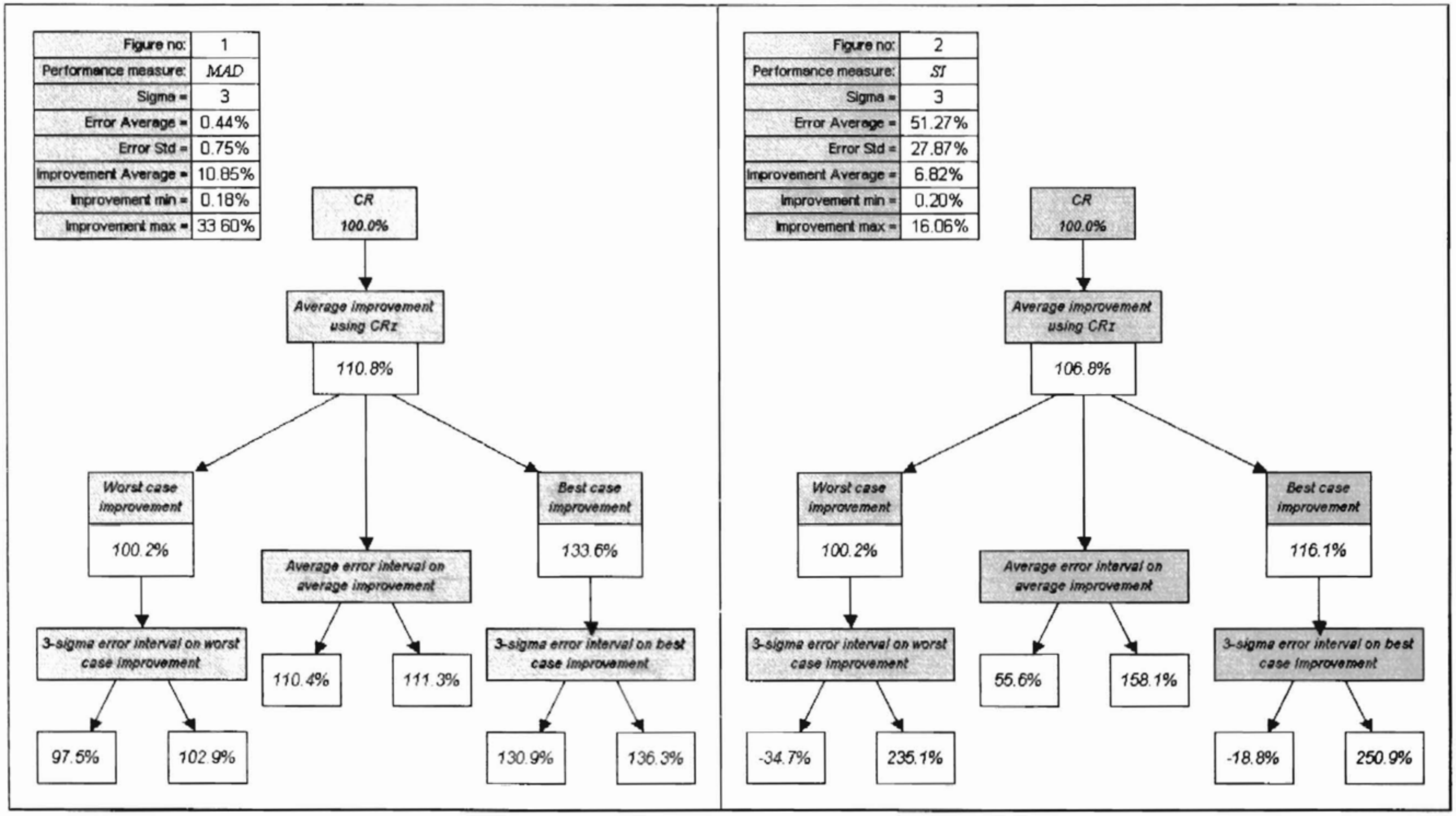

Figure 6 - Best case/worse case analysis for CRz improvement over CR, using *MAD* and *SI*

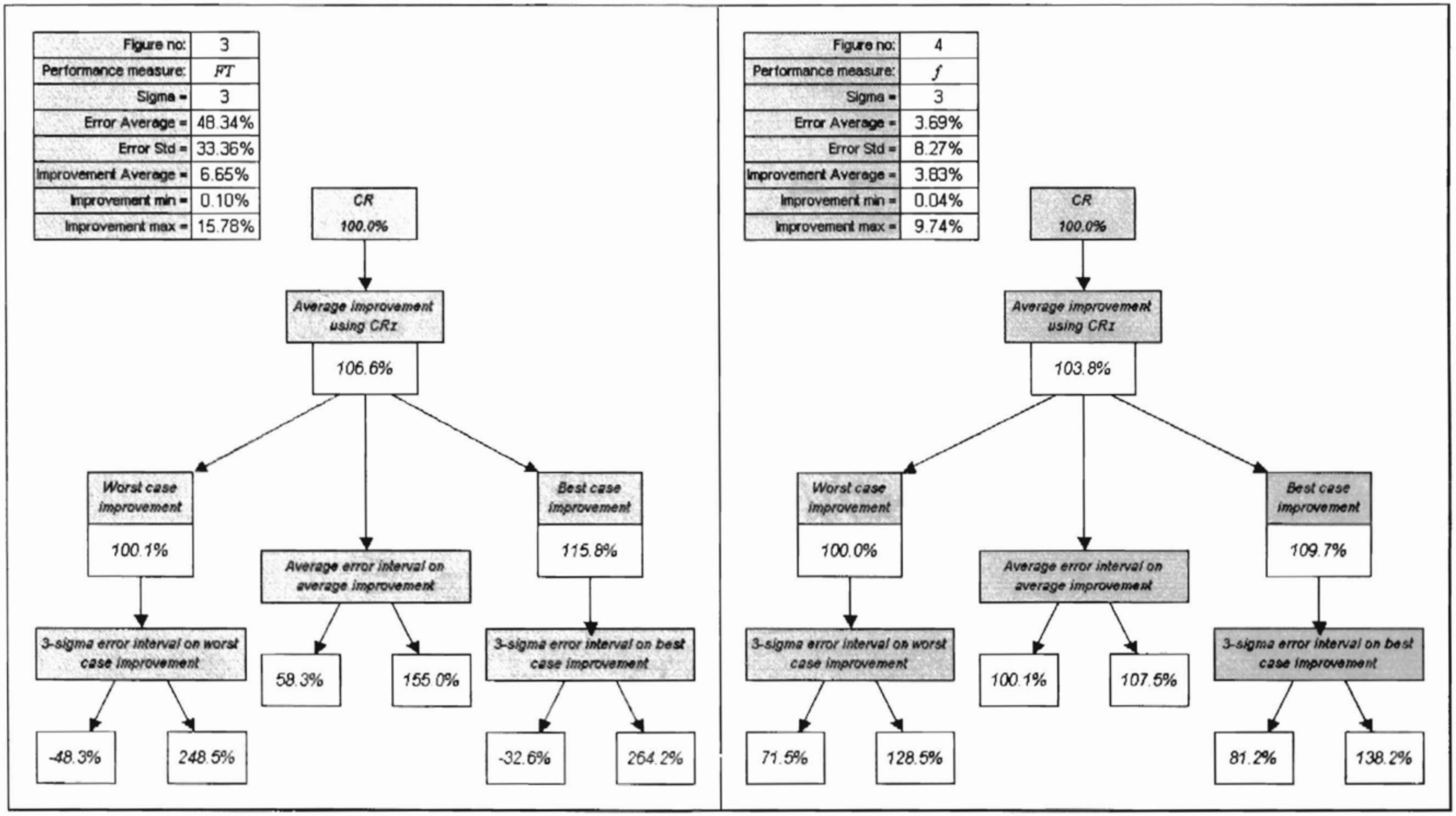

Figure 7 - Best case/worse case analysis for CRz improvement over CR, using *FT* and *f*

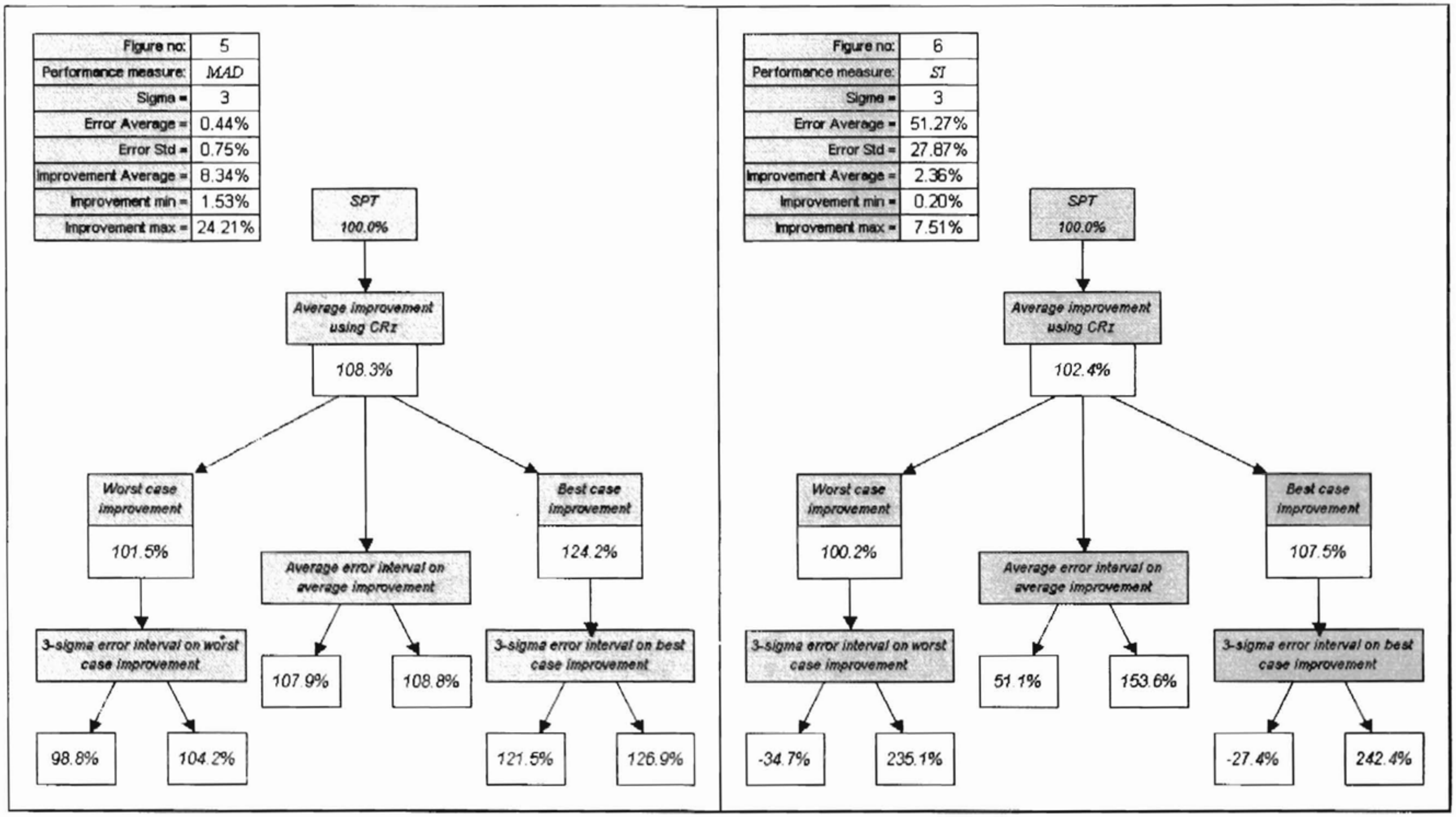

Figure 8 - Best case/worse case analysis for CRz improvement over SPT, using *MAD* and *SI*

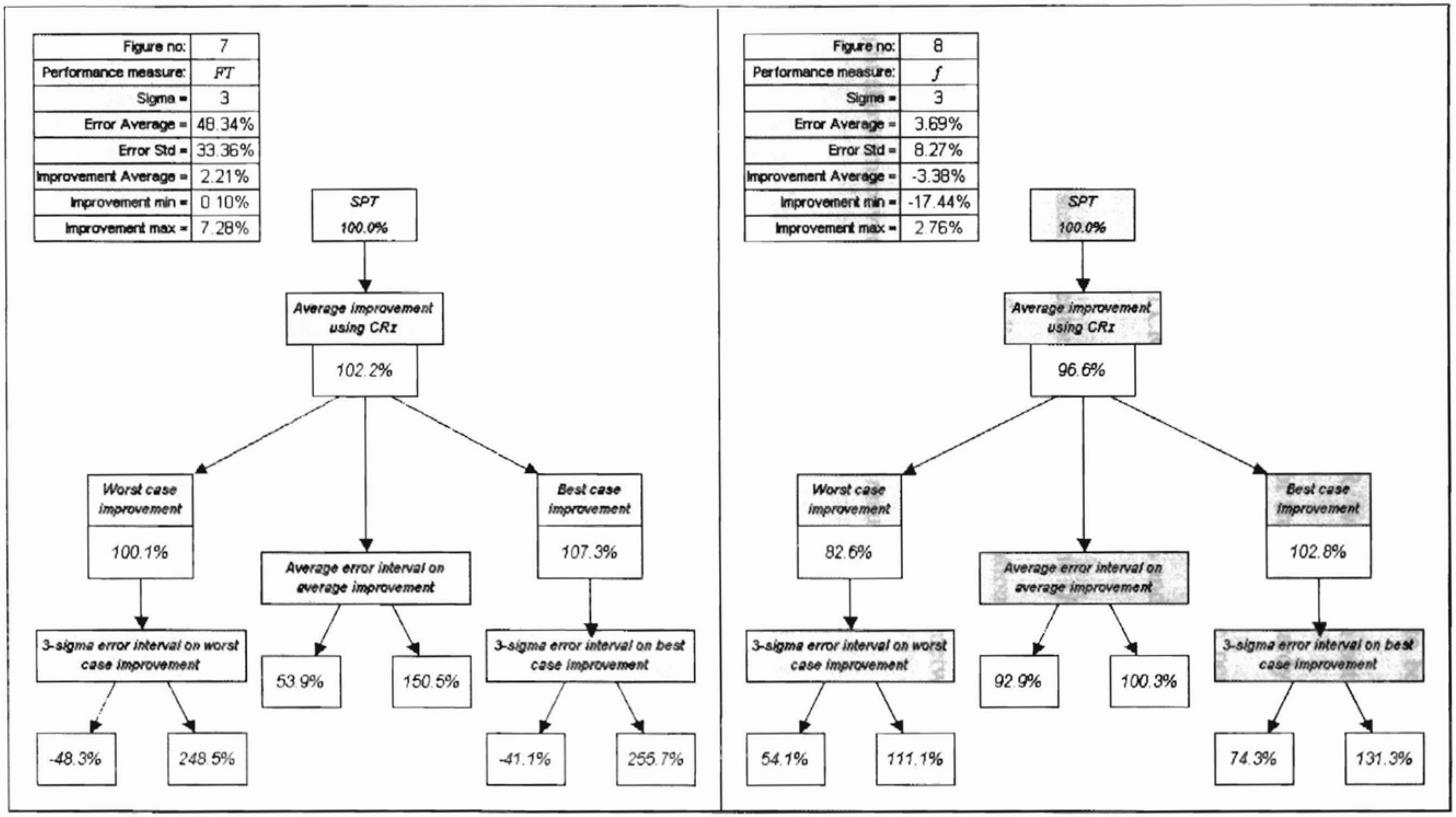

Figure 9 - Best case/worse case analysis for CRz improvement over SPT, using *FT* and *f*

It is obvious that using neural network approximation to determine z values for mean absolute deviation from due date (*MAD*) proved to be successful. Considering the two extreme cases, the performance of CR_z would be in between 98% and 136% of CR, and between 99% and 127% of SPT. As for fraction of tardy jobs (*f*) only limited success was achieved. Considering the two extreme cases, the performance of CR_z would be between 72% and 138% of CR, and between 54% and 131% of SPT.

As for system inventory (*SI*) and flow time (*FT*) the methodology did not predict the best z values successfully. Further investigation and research might uncover more information into the reasons for this result. Other neural networks designs or architectures might show improvement over the current results.

VI Conclusion

This chapter discusses the conclusions, findings, and insights of this research. Research contribution and future directions are also presented in this chapter.

VI.1 Research Findings and Insights

A significant finding of this research was the fact that selecting a sequencing rule in a particular job shop is directly influenced by the system configuration parameters that describe the studied job shop. Therefore, this research deployed this finding in predicting the best power factor (z) as a function of system configuration parameters. This finding complements earlier research findings which suggest that the selection of sequencing rules is affected by the performance measure used in the job shop studied. The indicated relationship between sequencing rules and system parameters suggests that different interaction levels do exist between sequencing rules and system parameters. Given the previous findings, job shop designers and researchers can now focus their improvement efforts where benefits can be maximized.

It should be noted that the methodology performance for predicting the best power factors when using f was quite acceptable except in the few cases where extremely high system utilization and tight due date allowance factor forced CR_z to fall behind SPT, hence the relatively low performance in the worst case scenarios. Still, when the methodology was deployed using *MAD*, it exceed both CR and SPT in all cases, which indicates that it should perform just as well (possibly better) under normal circumstances.

Another important finding was the fact that different performance measures have different sensitivity levels to errors in predicting z. This finding suggests that performance measures have different response sensitivity levels to system configuration parameters changes, which in turn means that performance measures have different dependency levels to specific job shop characteristics.

The above finding might be the reason behind the poor performance in the cases of *SI* and *FT*. It is quite possible that the sensitivity level in both cases was much lower than the error level in the proposed methodology; errors are inherently associated with approximation methods as well as in simulation based methods. Therefore, the interaction between performance measures and system parameters still exists, but may not be obvious due to the relatively high approximation error noise. Further investigation into this phenomenon may generate more insight into the problem.

As for the technical part of this research, a comparison between two neural network designs and a multiple regression model showed that regression does not hold much potential in such complicated research areas. The research also reinforced a well-known shortcoming to Radial Basis Neural Networks (RBNN) approximation method, which is its poor generalization over non-training cases.

Finally, the current research introduced a new robust methodology that allows the accommodation of certain types of disruption to the job shop system studied. For example, if the production plan required a system utilization increase, or the marketing department required tighter due dates, the new methodology accommodates the change by adjusting z to a new value that provides the best performance level in light of the new changes. In contrast, traditional sequencing rules and methodologies required the initiation of an analytical process that will determine the best sequencing rule, this is often an iterative and lengthy process which might affect the system robustness to external changes.

It should be noted that certain types of disruptions to the job shop system studied would require the methodology to be re-run to determine a new value for z. An example of such disruptions is the violation of the job shop definition (i.e. adding a machine or changing the route).

VI.2 Contribution

This research follows a different paradigm to solve an old problem, job shop scheduling. The new paradigm escapes the traditional method of selecting the best sequencing rule for a particular job shop by using a single sequencing rule that possesses the capability of adjusting to particular system conditions by the means of a factor. This research focuses on the determination of this factor using approximation methods.

Basically, this research introduced a methodology to facilitate the application of the Modified Critical Ratio Rule (CR_z). As a result, the best job shop performance could be achieved using a z factor determined by the system configuration parameters. This allows the system to dynamically accommodate external disruptions, simply by adjusting the z factor. Furthermore, the methodology can be used as a tool for exploring interaction, response sensitivity, and dependency levels between system parameters and performance measures. The exploration of such characteristics would lead to a deeper understanding of manufacturing science.

VI.3 Future Directions

This research combined different fields of knowledge for the sake of providing the proposed methodology. Because of that, there are multiple connected yet different ways to further advance this research in the future. Future directions in this research could be divided into two groups: further generalization of the methodology, and further investigation into the methodology. The following are some of the proposed future directions to further generalize the methodology.

1. This research considered only a limited set of system configuration parameters, and performance measures. Future research may investigate a more comprehensive list of system configuration parameters that might include job shop characteristics (i.e. number of machines, job routing) and workcenter-specific characteristics (i.e. number of jobs currently in queue, scrap). Other performance measures could also be included in future research to further generalize the proposed methodology. One aspect that could extend the methodology is to include job shop model specifications in the list of system configuration parameters in order to include a wider base for applying the proposed methodology.

2. Because of simulation execution time restrictions, this research limited the power factor (z) to a relatively limited range, which might exclude feasible solutions from the methodology. Future research should not exclude such solutions, if possible.

3. Neural networks could include multiple outputs in one design; this feature could be used to advance the proposed methodology so that more integrated models analysis is available. This design considers the effect of inputs on several outputs in the same model. Future research may include further modifications to the CR_z rule, or to even to apply the proposed methodology to new parameterized sequencing rules.

The following are some of the proposed future directions to further investigation in the methodology:

1. One possible use of the proposed methodology is to explore the effect of system configuration parameters on interaction, sensitivity, and dependency levels for different performance measures.

2. High error levels in predicting *SI* and *FT* may be the reason of poor performance of the methodology with regard to those two performance measures. It is quite possible that the response sensitivity level in both cases was much lower than the error level in the proposed methodology. Therefore, the interaction between performance measures and system parameters still exists, but are not obvious due to relatively high

approximation error noise. Further investigation into this matter may provide more insight into the problem.

Bibliography

Abu-Suleiman, A. (1998), "Job Shop Scheduling: A Quantified Sequencing Rule for Improving System Performance Under Diversified Operational Parameters," M.S. Thesis, School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK.

Baker, K.R. (1984), "Sequencing Rules and Due Date Assignments in a Job Shop," Management Science, 30, 9, 1093-1104.

Bauer, K.B. (1988), "Parallel Distributed Processing Models Using the Back-Propagation Rule for Studying Analytic and Holistic Modes of Processing in Category Learning," M.S. Thesis, Texas A&M University, College Station Texas.

Billings, S.A., H.B. Jamaluddin, S. Chen (1992), "Properties of Neural Networks with Applications to Modeling Non-linear Dynamic Systems," International Journal of Control 55, 1, 193-224.

Caudill, M. (1989), *Neural Networks Primer*, Miller Freeman, San Francisco, Ca.

Cook, D.F., R.E. Shannon (1992), "A Predictive Neural Network Modeling System for Manufacturing Process Parameters," International Journal of Production Research, 30, 7, 1537-1550.

Eilon, S., I.G. Chowdhury (1976), "Due Dates in Job Shop Scheduling," International Journal of Production Research, 14, 2, 223-237.

French, S. (1982), *Sequencing and Scheduling: An Introduction to The Mathematics of The Job-Shop*, First Edition, Halsted Press, New York, NY.

Hagan, M., H.B. Demuth, M. Beale (1996), *Neural Networks Design*, PWS Publishing, Boston, MA.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, Second Edition, Prentice Hall, Upper Side River, New Jersey.

Hinton, G.E. (1992), "How Neural Networks Learn from Experience," Scientific American, Special Issue: Mind and Brain, September 1992. 145-151.

Kanet, J.J., J.C. Hayya (1982), "Priority Dispatching with Operation Due Dates in a Job Shop," Journal of Operations Management, 2, 3, 167-174.

Kelton, W.D, R.P. Sadowski, D.A. Sadowski (1998), *Simulation with ARENA*, WCB McGraw-Hill, New York, NY.

Law, M., W. Kelton (1991), *Simulation Modeling and Analysis*, Second edition, McGraw-Hill, New York, NY.

Lee, Y.H., M. Pinedo (1992), "Scheduling Jobs on Parallel Machines with Sequence-Dependent Setup Times," Technical report, Columbia University, New York, NY.

Lee, Y.H., S. Kim (1993), "Neural Network Applications for Scheduling Jobs on Parallel Machines," Computers and Industrial Engineering, 25, 1-4, 227-230.

Leu, Bor-Yuh (1994), "An Investigation of Group Scheduling Heuristics in a Flow Shop Cellular System with Work Center Sharing for The Forbidden Early Shipment Environment," PhD. Dissertation, School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK.

MacKay, D. J. C. (1992), "Bayesian Interpolation," Neural Computation, 4, 3, 415-447.

Math Works (1999), *MatLab: The Language of Technical Computing*, The Math Works, Natick, MA.

Math Works (2000), *Neural Networks Toolbox for Use with MATLAB*, The Math Works, Natick, MA.

Neter, J., W. Wasserman, M.H. Kutner (1989), *Applied Linear Regression Models*, Second Edition, Irwin, Homewood, IL.

Philipoom, P.R., L.P. Rees, L. Wiegmann (1994), "Using Neural Networks to Determine Internally-Set Due-Date Assignments for Shop Scheduling," Decision Sciences, 25, 5/6, 825-851.

Rohleder, T.R., G.D. Scudder (1991), "Scheduling Rule Selection for The Forbidden Early Shipment Environment: A Comparison of Economic Objectives," International Journal of Production Research, 30, 1, 129-140.

Sabuncuoglu, I., T. Lejmi (1999), "Scheduling for Non Regular Performance Measure Under The Due Window Approach," Omega: The International Journal of Management Science, 27, 555-568.

SAS (1990), *SAS/STAT user's guide*, SAS Institute Inc, Cary, NC.

Vollmann, T.E., Berry, W.L., Whybark, D.C. (1997), *Manufacturing Planning and Control Systems*. Irwin/McGraw-Hill, New York, NY.

Widjaja, P. (1997), "A Simulation Evaluation of Scheduling Heuristics in a Dynamic Flow Shop Environment," M.S. Thesis, School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK.

Appendices

Appendix A – Simulation Model Description

The simulation model used in this research was built using Arena software. While Arena is only a graphical representation of the actual SIMAN code, this Appendix will include both of the graphical model and the SIMAN code behind the model. In addition, a discussion of the model is presented.

The simulation model is divided into two groups of blocks: main control blocks (Figure 10), and job shop model blocks (Figure 11). The main control blocks are responsible for the major simulation events such as collecting statistics, reading inputs, writing outputs, defining variables, defining expressions, and starting and terminating replications.

The job shop model blocks are responsible for representing the actual job shop model. Entities (jobs) arrive at the ARRIVE block with a pre-determined arrival rate. Attributes are assigned to entities in the two ASSING blocks, the CR_Z value will be assigned in the second ASSIGN block for later use. System variables are also updated in these blocks. Next, entities are randomly routed to one of the SERVER blocks using the CHOOSE block, in each SERVER block entities are delayed for the assigned processing time. When the delay is over, entities go through an ASSING block to update system variables and entity attributes. Next, a CHOOSE block determines weather the entity has finished the required number of processes, if not, a new machine is randomly assigned as the next machine, taking into consideration the restrictions defined in the actual job shop model. If the entity has finished all required processes, it is routed to a statistics collecting stage. The statistics collecting stage starts with a CHOOSE block to determine if the entity is early or tardy, in either case earliness or tardiness in addition to absolute deviation from due date are recorded. In the case of being early, the entity waits until it's due date using a DELAY block. Finally, flow time statistics are recorded and the number of finished jobs, fraction of tardy jobs, and system inventory variables are updated using the TALLY, COUNT, and ASSIGN blocks. Entities are disposed of using the DEPART block.

It's important to note that entities are prioritized in the server block queue according to their assigned CR_z values; lower values have higher priority. Calculating CR_z values uses some of the predefined expressions found in the main control blocks. The CR_z value of an entity is recalculated each time the entity loops back to the next machine.

In the following pages pictures of the graphical models are represented. In addition, the contents of the expression file and the SIMAN code are presented in this Appendix.

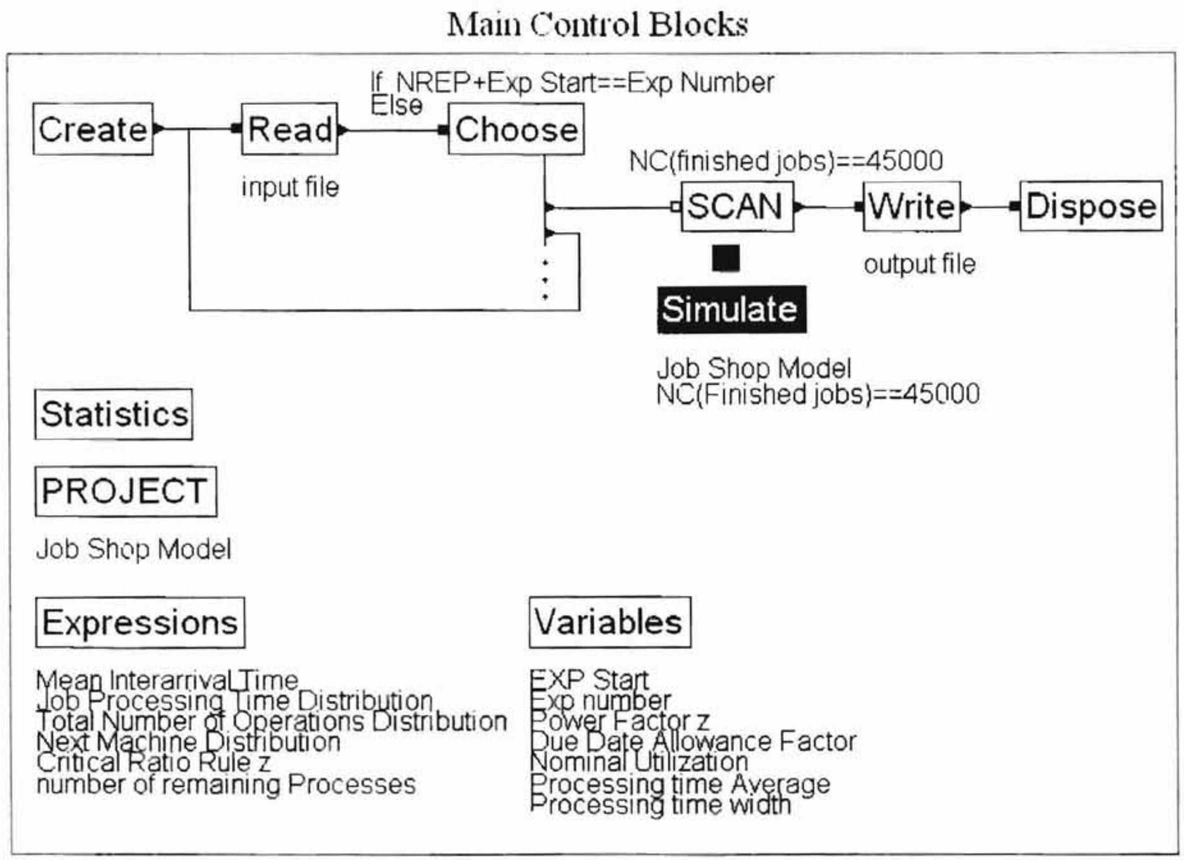

Figure 10 - Arena graphical model: main control blocks

Job Shop Model

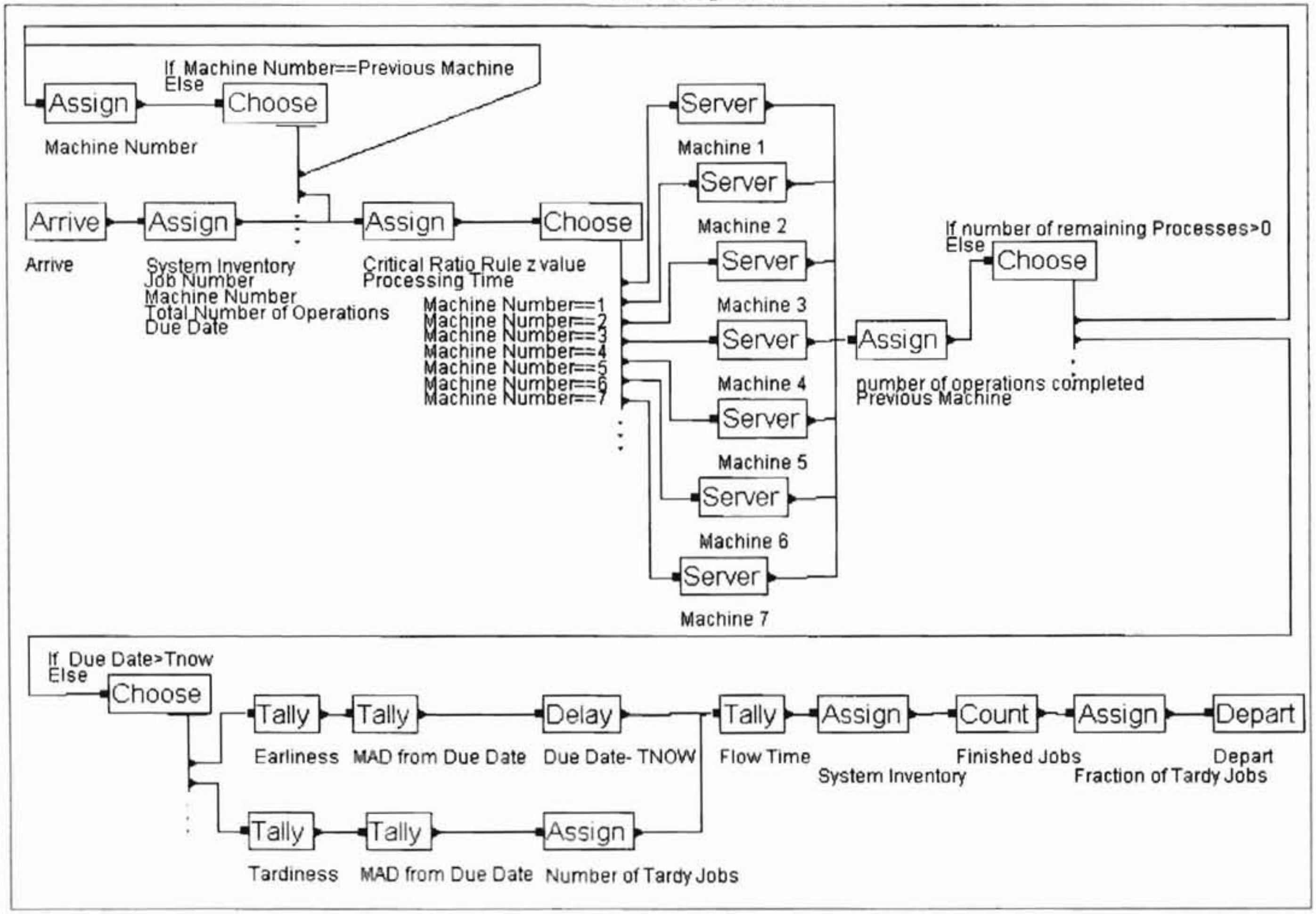

Figure 11 - Arena graphical model: job shop model blocks

The following is the experiment file:

```
PROJECT,      Job Shop Model,Sinan Salman,03/30/2001,No;

ATTRIBUTES:   Job Number:
              Previous Machine:
              number of operations completed:
              Processing Time:
              Critical Ratio Rule z value:
              Due Date:
              Arrival Time:
              QueueTime:
              Machine Number:
              Total Number of Operations;

FILES:        output file,"output.txt",(),Free Format:
              input file,"input.txt",(),Free Format,Dispose;

VARIABLES:    Power Factor z:
              System Inventory:
              Number of Tardy Jobs:
              Processing time Average:
              Fraction of Tardy Jobs:
              Processing time width:
              Due Date Allowance Factor:
              Nominal Utilization:
              Exp number:
              EXP Start,0;

QUEUES:       Machine 4_R_Q,LVF(Critical Ratio Rule z value):
              Machine 5_R_Q,LVF(Critical Ratio Rule z value):
              Machine 6_R_Q,LVF(Critical Ratio Rule z value):
              Machine 7_R_Q,LVF(Critical Ratio Rule z value):
              Machine 1_R_Q,LVF(Critical Ratio Rule z value):
              Machine 2_R_Q,LVF(Critical Ratio Rule z value):
              Machine 3_R_Q,LVF(Critical Ratio Rule z value);

PICTURES:     Default;

RESOURCES:    Machine 3_R,Capacity(1,),-,Stationary:
              Machine 4_R,Capacity(1,),-,Stationary:
              Machine 5_R,Capacity(1,),-,Stationary:
              Machine 6_R,Capacity(1,),-,Stationary:
              Machine 7_R,Capacity(1,),-,Stationary:
              Machine 1_R,Capacity(1,),-,Stationary:
              Machine 2_R,Capacity(1,),-,Stationary;

STATIONS:     Machine 3:
              Machine 4:
              Machine 5:
              Machine 6:
              Machine 7:
              Depart:
              Arrive:
              Machine 1:
              Machine 2;

COUNTERS:     Finished Jobs,,Replicate:
              Number of Jobs,,Replicate;

TALLIES:      Machine 4_R_Q Queue Time:
              Tardiness:
              MAD from Due Date:
              Machine 5_R_Q Queue Time:
              Machine 1_R_Q Queue Time:
              Machine 6_R_Q Queue Time:
              Machine 2_R_Q Queue Time:
              Flow Time:
```

```
               Machine 7_R_Q Queue Time:
               Earliness:
               Machine 3_R_Q Queue Time;

DSTATS:        MR(Machine 2_R),Machine 2_R Available:
               System Inventory,System Inventory output:
               MR(Machine 1_R),Machine 1_R Available:
               NQ(Machine 7_R_Q),# in Machine 7_R_Q:
               NQ(Machine 6_R_Q),# in Machine 6_R_Q:
               NQ(Machine 5_R_Q),# in Machine 5_R_Q:
               NQ(Machine 4_R_Q),# in Machine 4_R_Q:
               NR(Machine 7_R),Machine 7_R Busy:
               NQ(Machine 3_R_Q),# in Machine 3_R_Q:
               NR(Machine 6_R),Machine 6_R Busy:
               NQ(Machine 2_R_Q),# in Machine 2_R_Q:
               NR(Machine 5_R),Machine 5_R Busy:
               NR(Machine 4_R),Machine 4_R Busy:
               NQ(Machine 1_R_Q),# in Machine 1_R_Q:
               NR(Machine 3_R),Machine 3_R Busy:
               Fraction of Tardy Jobs,Fraction of Tardy Jobs output:
               NR(Machine 2_R),Machine 2_R Busy:
               NR(Machine 1_R),Machine 1_R Busy:
               MR(Machine 7_R),Machine 7_R Available:
               MR(Machine 6_R),Machine 6_R Available:
               MR(Machine 5_R),Machine 5_R Available:
               MR(Machine 4_R),Machine 4_R Available:
               MR(Machine 3_R),Machine 3_R Available;

REPLICATE,     61425,0.0,,Yes,Yes,15000,NC(Finished jobs)==45000;

EXPRESSIONS:   Job Processing Time Distribution,
               UNIF(Processing time average-(Processing time width/2),Processing time
average+(Processing time width/2),1):
               Next Machine Distribution,AINT(UNIF(1,8,3)):
               Total Number of Operations Distribution,AINT(UNIF(3,8,2)):
               Mean Interarrival Time,1/(1.4 * Nominal Utilization / Processing time
average):
               Critical Ratio Rule z,
               (Due Date - TNOW) / (Processing time average* number of remaining
Processes)**Power Factor z:
               number of remaining Processes,Total Number of Operations - Number of
Operations Completed;
```

The following is the SIMAN code generated by Arena:

```
;
;
;     Model statements for module:  Arrive 1
;

77$           CREATE,        1,.0002:EXPO(Mean Interarrival Time,4):MARK(Arrival Time);

38$           STATION,       Arrive;
86$           TRACE,         -1,"-Arrived to system at station Arrive\n":;
41$           ASSIGN:        Picture=Default;
62$           DELAY:         0.;
101$          COUNT:         Number of Jobs,1;
90$           TRACE,         -1,"-Transferred to next module\n"::NEXT(0$);


;
;
;     Model statements for module:  Assign 1
;
0$            TRACE,         -1,"-Making assignments\n":;
117$          ASSIGN:        System Inventory=System Inventory+1:
                             Job Number=NC(Number of Jobs):
                             Machine Number=Next Machine Distribution:
                             Total Number of Operations=Total Number of Operations
Distribution:
                             Due Date=TNOW + (Due Date Allowance Factor * Total Number of
Operations * Processing Time Average)
                             :NEXT(17$);


;
;
;     Model statements for module:  Assign 11
;
17$           TRACE,         -1,"-Making assignments\n":;
118$          ASSIGN:        Critical Ratio Rule z value=Critical Ratio Rule z:
                             Processing Time=Job Processing Time Distribution:NEXT(1$);


;
;
;     Model statements for module:  Choose 2
;
1$            TRACE,         -1,"-Choosing from 7 options\n":;
119$          BRANCH,        1:
                             If,Machine Number==1,2$,Yes:
                             If,Machine Number==2,4$,Yes:
                             If,Machine Number==3,6$,Yes:
                             If,Machine Number==4,8$,Yes:
                             If,Machine Number==5,11$,Yes:
                             If,Machine Number==6,13$,Yes:
                             If,Machine Number==7,15$,Yes;

;
;
;     Model statements for module:  Server 1
;

2$            STATION,       Machine 1;
196$          TRACE,         -1,"-Arrived to station Machine 1\n":;
159$          DELAY:         0.;
203$          TRACE,         -1,"-Waiting for resource Machine 1_R\n":;
120$          QUEUE,         Machine 1_R_Q:MARK(QueueTime);
121$          SEIZE,         1:
                             Machine 1_R,1;
```

```
230$          BRANCH,        1:
                             If,RTYP(Machine 1_R).eq.2,231$,Yes:
                             If,RTYP(Machine 1_R).eq.1,133$,Yes;
231$          MOVE:          Machine 1_R,Machine 1;
133$          TALLY:         Machine 1_R_Q Queue Time,INT(QueueTime),1;
240$          DELAY:         0.0;
              TRACE,         -1,"-Delay for processing time Processing Time\n":;
122$          DELAY:         Processing Time;
204$          TRACE,         -1,"-Releasing resource\n":;
123$          RELEASE:       Machine 1_R,1;
187$          DELAY:         0.;
210$          TRACE,         -1,"-Transferred to next module\n"::NEXT(10$);


;
;
;     Model statements for module:  Assign 6
;
10$           TRACE,         -1,"-Making assignments\n":;
241$          ASSIGN:        number of operations completed=Number of Operations
Completed+1:
                             Previous Machine=Machine Number:NEXT(30$);


;
;
;     Model statements for module:  Choose 10
;
30$           TRACE,         -1,"-Choosing from 2 options\n":;
242$          BRANCH,        1:
                             If,number of remaining Processes>0,32$,Yes:
                             Else,29$,Yes;

;
;
;     Model statements for module:  Assign 14
;
32$           TRACE,         -1,"-Making assignments\n":;
243$          ASSIGN:        Machine Number=Next Machine Distribution:NEXT(31$);


;
;
;     Model statements for module:  Choose 9
;
31$           TRACE,         -1,"-Choosing from 2 options\n":;
244$          BRANCH,        1:
                             If,Machine Number==Previous Machine,32$,Yes:
                             Else,17$,Yes;

;
;
;     Model statements for module:  Choose 6
;
29$           TRACE,         -1,"-Choosing from 2 options\n":;
245$          BRANCH,        1:
                             If,Due Date>Tnow,27$,Yes:
                             Else,28$,Yes;

;
;
;     Model statements for module:  Tally 3
;
27$           TRACE,         -1,"-Updating Tally Earliness \n":;
246$          TALLY:         Earliness,Due Date- TNOW,1:NEXT(34$);


;
;
;     Model statements for module:  Tally 5
```

```
;
34$          TRACE,          -1,"-Updating Tally MAD from Due Date \n":;
248$         TALLY:          MAD from Due Date,Due Date- TNOW,1:NEXT(33$);


;
;
;     Model statements for module:  Delay 1
;
33$          TRACE,          -1,"-Delaying for time Due Date- TNOW\n":;
250$         DELAY:          Due Date- TNOW:NEXT(25$);


;
;
;     Model statements for module:  Tally 2
;
25$          TRACE,          -1,"-Updating Tally Flow Time \n":;
251$         TALLY:          Flow Time,Interval(Arrival Time),1:NEXT(19$);


;
;
;     Model statements for module:  Assign 12
;
19$          TRACE,          -1,"-Making assignments\n":;
253$         ASSIGN:         System Inventory=System Inventory-1:NEXT(24$);


;
;
;     Model statements for module:  Count 2
;
24$          TRACE,          -1,"-Updating counter Finished Jobs \n":;
254$         COUNT:          Finished Jobs,1:NEXT(37$);


;
;
;     Model statements for module:  Assign 18
;
37$          TRACE,          -1,"-Making assignments\n":;
256$         ASSIGN:         Fraction of Tardy Jobs=Number of tardy jobs/NC(Finished
Jobs):NEXT(18$);


;
;
;     Model statements for module:  Depart 1
;

18$          STATION,        Depart;
287$         TRACE,          -1,"-Arrived to station Depart\n":;
257$         DELAY:          0.;
294$         TRACE,          -1,"-Disposing entity\n":;
286$         DISPOSE;


;
;
;     Model statements for module:  Tally 4
;
28$          TRACE,          -1,"-Updating Tally Tardiness \n":;
296$         TALLY:          Tardiness,TNOW - Due Date,1:NEXT(35$);


;
;
;     Model statements for module:  Tally 6
;
```

```
35$          TRACE,        -1,"-Updating Tally MAD from Due Date \n":;
298$         TALLY:        MAD from Due Date,TNOW - Due Date,1:NEXT(36$);


;
;
;     Model statements for module:  Assign 16
;
36$          TRACE,        -1,"-Making assignments\n":;
300$         ASSIGN:       Number of Tardy Jobs=Number of Tardy Jobs+1:NEXT(25$);


;
;
;     Model statements for module:  Server 2
;

4$           STATION,      Machine 2;
377$         TRACE,        -1,"-Arrived to station Machine 2\n":;
340$         DELAY:        0.;
384$         TRACE,        -1,"-Waiting for resource Machine 2_R\n":;
301$         QUEUE,        Machine 2_R_Q:MARK(QueueTime);
302$         SEIZE,        1:
                           Machine 2_R,1;
411$         BRANCH,       1:
                           If,RTYP(Machine 2_R).eq.2,412$,Yes:
                           If,RTYP(Machine 2_R).eq.1,314$,Yes;
412$         MOVE:         Machine 2_R,Machine 2;
314$         TALLY:        Machine 2_R_Q Queue Time,INT(QueueTime),1;
421$         DELAY:        0.0;
             TRACE,        -1,"-Delay for processing time Processing Time\n":;
303$         DELAY:        Processing Time;
385$         TRACE,        -1,"-Releasing resource\n":;
304$         RELEASE:      Machine 2_R,1;
368$         DELAY:        0.;
391$         TRACE,        -1,"-Transferred to next module\n"::NEXT(10$);


;
;
;     Model statements for module:  Server 3
;

6$           STATION,      Machine 3;
498$         TRACE,        -1,"-Arrived to station Machine 3\n":;
461$         DELAY:        0.;
505$         TRACE,        -1,"-Waiting for resource Machine 3_R\n":;
422$         QUEUE,        Machine 3_R_Q:MARK(QueueTime);
423$         SEIZE,        1:
                           Machine 3_R,1;
532$         BRANCH,       1:
                           If,RTYP(Machine 3_R).eq.2,533$,Yes:
                           If,RTYP(Machine 3_R).eq.1,435$,Yes;
533$         MOVE:         Machine 3_R,Machine 3;
435$         TALLY:        Machine 3_R_Q Queue Time,INT(QueueTime),1;
542$         DELAY:        0.0;
             TRACE,        -1,"-Delay for processing time processing time\n":;
424$         DELAY:        processing time;
506$         TRACE,        -1,"-Releasing resource\n":;
425$         RELEASE:      Machine 3_R,1;
489$         DELAY:        0.;
512$         TRACE,        -1,"-Transferred to next module\n"::NEXT(10$);


;
;
;     Model statements for module:  Server 4
;

8$           STATION,      Machine 4;
```

```
619$          TRACE,         -1,"-Arrived to station Machine 4\n":;
582$          DELAY:         0.;
626$          TRACE,         -1,"-Waiting for resource Machine 4_R\n":;
543$          QUEUE,         Machine 4_R_Q:MARK(QueueTime);
544$          SEIZE,         1:
                             Machine 4_R,1;
653$          BRANCH,        1:
                             If,RTYP(Machine 4_R).eq.2,654$,Yes:
                             If,RTYP(Machine 4_R).eq.1,556$,Yes;
654$          MOVE:          Machine 4_R,Machine 4;
556$          TALLY:         Machine 4_R_Q Queue Time,INT(QueueTime),1;
663$          DELAY:         0.0;
              TRACE,         -1,"-Delay for processing time Processing Time\n":;
545$          DELAY:         Processing Time;
627$          TRACE,         -1,"-Releasing resource\n":;
546$          RELEASE:       Machine 4_R,1;
610$          DELAY:         0.;
633$          TRACE,         -1,"-Transferred to next module\n"::NEXT(10$);


;
;
;     Model statements for module:  Server 5
;

11$           STATION,       Machine 5;
740$          TRACE,         -1,"-Arrived to station Machine 5\n":;
703$          DELAY:         0.;
747$          TRACE,         -1,"-Waiting for resource Machine 5_R\n":;
664$          QUEUE,         Machine 5_R_Q:MARK(QueueTime);
665$          SEIZE,         1:
                             Machine 5_R,1;
774$          BRANCH,        1:
                             If,RTYP(Machine 5_R).eq.2,775$,Yes:
                             If,RTYP(Machine 5_R).eq.1,677$,Yes;
775$          MOVE:          Machine 5_R,Machine 5;
677$          TALLY:         Machine 5_R_Q Queue Time,INT(QueueTime),1;
784$          DELAY:         0.0;
              TRACE,         -1,"-Delay for processing time Processing Time\n":;
666$          DELAY:         Processing Time;
748$          TRACE,         -1,"-Releasing resource\n":;
667$          RELEASE:       Machine 5_R,1;
731$          DELAY:         0.;
754$          TRACE,         -1,"-Transferred to next module\n"::NEXT(10$);


;
;
;     Model statements for module:  Server 6
;

13$           STATION,       Machine 6;
861$          TRACE,         -1,"-Arrived to station Machine 6\n":;
824$          DELAY:         0.;
868$          TRACE,         -1,"-Waiting for resource Machine 6_R\n":;
785$          QUEUE,         Machine 6_R_Q:MARK(QueueTime);
786$          SEIZE,         1:
                             Machine 6_R,1;
895$          BRANCH,        1:
                             If,RTYP(Machine 6_R).eq.2,896$,Yes:
                             If,RTYP(Machine 6_R).eq.1,798$,Yes;
896$          MOVE:          Machine 6_R,Machine 6;
798$          TALLY:         Machine 6_R_Q Queue Time,INT(QueueTime),1;
905$          DELAY:         0.0;
              TRACE,         -1,"-Delay for processing time Processing Time\n":;
787$          DELAY:         Processing Time;
869$          TRACE,         -1,"-Releasing resource\n":;
788$          RELEASE:       Machine 6_R,1;
852$          DELAY:         0.;
875$          TRACE,         -1,"-Transferred to next module\n"::NEXT(10$);
```

```
;
;
;     Model statements for module:  Server 7
;

15$           STATION,       Machine 7;
982$          TRACE,         -1,"-Arrived to station Machine 7\n":;
945$          DELAY:         0.;
989$          TRACE,         -1,"-Waiting for resource Machine 7_R\n":;
906$          QUEUE,         Machine 7_R_Q:MARK(QueueTime);
907$          SEIZE,         1:
                             Machine 7_R,1;
1016$         BRANCH,        1:
                             If,RTYP(Machine 7_R).eq.2,1017$,Yes:
                             If,RTYP(Machine 7_R).eq.1,919$,Yes;
1017$         MOVE:          Machine 7_R,Machine 7;
919$          TALLY:         Machine 7_R_Q Queue Time,INT(QueueTime),1;
1026$         DELAY:         0.0;
              TRACE,         -1,"-Delay for processing time Processing Time\n":;
908$          DELAY:         Processing Time;
990$          TRACE,         -1,"-Releasing resource\n":;
909$          RELEASE:       Machine 7_R,1;
973$          DELAY:         0.;
996$          TRACE,         -1,"-Transferred to next module\n"::NEXT(10$);


;
;
;     Model statements for module:  Create 1
;

1027$         CREATE,        1,.0001:,1;
1034$         TRACE,         -1,"-Entity Created\n":;
1031$         ASSIGN:        Picture=Default:NEXT(20$);


;
;
;     Model statements for module:  Read 1
;
20$           TRACE,         -1,"-Reading from input file \n":;
1035$         READ,          input file:
                             Exp number,
                             Processing time Average,
                             Processing time width,
                             Due Date Allowance Factor,
                             Nominal Utilization,
                             Power Factor z:NEXT(21$);


;
;
;     Model statements for module:  Choose 5
;
21$           TRACE,         -1,"-Choosing from 2 options\n":;
1036$         BRANCH,        1:
                             If,NREP+Exp Start==Exp Number,26$,Yes:
                             Else,20$,Yes;
26$           SCAN:          NC(finished jobs)==45000:NEXT(23$);


;
;
;     Model statements for module:  Write 1
;
23$           TRACE,         -1,"-Writing to File output file\n":;
1037$         WRITE,         output file,
```

```
"(i5,1x,f5.2,1x,f5.3,1x,i2,1x,f5.3,1x,f7.3,1x,f8.4,1x,f8.4,1x,f8.4,1x,f8.4,1x,f8.4,1x,f6.
4)":
                              NREP+exp start,
                              Processing time Average,
                              Processing time width,
                              Due Date Allowance Factor,
                              Nominal Utilization,
                              Power Factor z,
                              TAVG(Earliness),
                              TAVG(Tardiness),
                              TAVG(MAD from Due Date),
                              DAVG(System inventory output),
                              TAVG(Flow Time),
                              Fraction of Tardy Jobs:NEXT(22$);


;
;
;     Model statements for module:  Dispose 1
;
22$           TRACE,          -1,"-Disposing entity\n"::
1038$         DISPOSE;
```

Appendix B - Verification of The Simulation Model

In this Appendix the full trace record is presented, this trace record ensures that simulation variables, entity's parameters, entity's routes, and stations behaviors are accurate representation of the actual job shop model. The following is a list of the system configuration parameters used in this trace:

- Due date allowance factor (k) = 3,
- Nominal system utilization (u) = 85%,
- Processing time distribution average (Pa) = 5,
- Processing time distribution interval width (Pw) = 3,
- CR_z Power factor (z) = -1.

The following is a trace of a tardy job throughout the simulation model:

```
Time: 1750.94  Entity: 31
   1 75$            CREATE
                              ARRIVAL TIME set to 1750.94
                              Next creation scheduled at time 1763.19
   2 36$            STATION
                              Entity 31 entered station ARRIVE
   3 84$            TRACE
   4 39$            ASSIGN
                              Entity 31 picture changed to DEFAULT
   5 60$            DELAY
                              Delayed by 0.0 until time 1750.94
   6 99$            COUNT
                              Counter NUMBER OF JOBS incremented by 1 to 438
   7 88$            TRACE
   8 0$             TRACE
   9 115$           ASSIGN
                              SYSTEM INVENTORY set to 24.0
                              JOB NUMBER set to 438.0
                              MACHINE NUMBER set to 4.0
                              TOTAL NUMBER OF OPERATIONS set to 4.0
                              DUE DATE set to 1810.94
  10 17$            TRACE
  11 116$           ASSIGN
                              CRITICAL RATIO RULE Z VALUE set to 1200.0
                              PROCESSING TIME set to 5.38175
  12 1$             TRACE
  13 117$           BRANCH
                              Selecting at most 1 of 7 branches
                              IF: Branch not selected
                              IF: Branch not selected
                              IF: Branch not selected
                              IF: Entity 31 sent to 8$
  95 8$             STATION
```

```
                                 Entity 31 entered station MACHINE 4
  96 616$            TRACE
  97 579$            DELAY
                                 Delayed by 0.0 until time 1750.94
  98 623$            TRACE
  99 540$            QUEUE
                                 QUEUETIME set to 1750.94
                                 Entity 31 sent to next block
 100 541$            SEIZE
                                 Entity 31 added to queue MACHINE 4_R_Q at rank 2
Time: 1782.9  Entity: 31
 101 650$            BRANCH
                                 Selecting at most 1 of 2 branches
                                 IF: Branch not selected
                                 IF: Entity 31 sent to 553$
 103 553$            TALLY
                                 Tally MACHINE 4_R_Q QUEUE TIME recorded 31.9564
 104 660$            DELAY
                                 Delayed by 0.0 until time 1782.9
 105                 TRACE
 106 542$            DELAY
                                 Delayed by 5.38175 until time 1788.28
Time: 1788.28  Entity: 31
 107 624$            TRACE
 108 543$            RELEASE
                                 MACHINE 4_R available increased by 1 to 1
                                 Entity 8 removed from queue MACHINE 4_R_Q
                                 Resource allocated to entity 8
                                 Seized 1 unit(s) of resource MACHINE 4_R
 109 607$            DELAY
                                 Delayed by 0.0 until time 1788.28
 110 630$            TRACE
                                 Entity transferred to block 10$
  30 10$             TRACE
  31 239$            ASSIGN
                                 NUMBER OF OPERATIONS COMPLETED set to 1.0
                                 PREVIOUS MACHINE set to 4.0
  32 29$             TRACE
  33 240$            BRANCH
                                 Selecting at most 1 of 2 branches
                                 IF: Entity 31 sent to 31$
  34 31$             TRACE
  35 241$            ASSIGN
                                 MACHINE NUMBER set to 2.0
  36 30$             TRACE
  37 242$            BRANCH
                                 Selecting at most 1 of 2 branches
                                 IF: Branch not selected
                                 ELSE: Entity 31 sent to 17$
  10 17$             TRACE
  11 116$            ASSIGN
                                 CRITICAL RATIO RULE Z VALUE set to 339.928
                                 PROCESSING TIME set to 5.66441
  12 1$              TRACE
  13 117$            BRANCH
                                 Selecting at most 1 of 7 branches
                                 IF: Branch not selected
                                 IF: Entity 31 sent to 4$
  63 4$              STATION
                                 Entity 31 entered station MACHINE 2
  64 374$            TRACE
  65 337$            DELAY
                                 Delayed by 0.0 until time 1788.28
  66 381$            TRACE
  67 298$            QUEUE
                                 QUEUETIME set to 1788.28
                                 Entity 31 sent to next block
  68 299$            SEIZE
                                 Could not seize resource MACHINE 2_R
                                 Entity 31 added to queue MACHINE 2_R_Q at rank 1
```

```
Time: 1796.38  Entity: 31
   69 408$              BRANCH
                                  Selecting at most 1 of 2 branches
                                  IF: Branch not selected
                                  IF: Entity 31 sent to 311$
   71 311$              TALLY
                                  Tally MACHINE 2_R_Q QUEUE TIME recorded 8.10611
   72 418$              DELAY
                                  Delayed by 0.0 until time 1796.38
   73                   TRACE
   74 300$              DELAY
                                  Delayed by 5.66441 until time 1802.05
Time: 1802.05  Entity: 31
   75 382$              TRACE
   76 301$              RELEASE
                                  MACHINE 2_R available increased by 1 to 1
                                  Entity 39 removed from queue MACHINE 2_R_Q
                                  Resource allocated to entity 39
                                  Seized 1 unit(s) of resource MACHINE 2_R
   77 365$              DELAY
                                  Delayed by 0.0 until time 1802.05
   78 388$              TRACE
                                  Entity transferred to block 10$
   30 10$               TRACE
   31 239$              ASSIGN
                                  NUMBER OF OPERATIONS COMPLETED set to 2.0
                                  PREVIOUS MACHINE set to 2.0
   32 29$               TRACE
   33 240$              BRANCH
                                  Selecting at most 1 of 2 branches
                                  IF: Entity 31 sent to 31$
   34 31$               TRACE
   35 241$              ASSIGN
                                  MACHINE NUMBER set to 1.0
   36 30$               TRACE
   37 242$              BRANCH
                                  Selecting at most 1 of 2 branches
                                  IF: Branch not selected
                                  ELSE: Entity 31 sent to 17$
   10 17$               TRACE
   11 116$              ASSIGN
                                  CRITICAL RATIO RULE Z VALUE set to 88.9138
                                  PROCESSING TIME set to 4.96332
   12 1$                TRACE
   13 117$              BRANCH
                                  Selecting at most 1 of 7 branches
                                  IF: Entity 31 sent to 2$
   14 2$                STATION
                                  Entity 31 entered station MACHINE 1
   15 194$              TRACE
   16 157$              DELAY
                                  Delayed by 0.0 until time 1802.05
   17 201$              TRACE
   18 118$              QUEUE
                                  QUEUETIME set to 1802.05
                                  Entity 31 sent to next block
   19 119$              SEIZE
                                  Could not seize resource MACHINE 1_R
                                  Entity 31 added to queue MACHINE 1_R_Q at rank 1
Time: 1803.97  Entity: 31
   20 228$              BRANCH
                                  Selecting at most 1 of 2 branches
                                  IF: Branch not selected
                                  IF: Entity 31 sent to 131$
   22 131$              TALLY
                                  Tally MACHINE 1_R_Q QUEUE TIME recorded 1.92171
   23 238$              DELAY
                                  Delayed by 0.0 until time 1803.97
   24                   TRACE
   25 120$              DELAY
```

```
                                        Delayed by 4.96332 until time 1808.93
Time: 1808.93  Entity: 31
   26 202$              TRACE
   27 121$              RELEASE
                                        MACHINE 1_R available increased by 1 to 1
   28 185$              DELAY
                                        Delayed by 0.0 until time 1808.93
   29 208$              TRACE
   30 10$               TRACE
   31 239$              ASSIGN
                                        NUMBER OF OPERATIONS COMPLETED set to 3.0
                                        PREVIOUS MACHINE set to 1.0
   32 29$               TRACE
   33 240$              BRANCH
                                        Selecting at most 1 of 2 branches
                                        IF: Entity 31 sent to 31$
   34 31$               TRACE
   35 241$              ASSIGN
                                        MACHINE NUMBER set to 4.0
   36 30$               TRACE
   37 242$              BRANCH
                                        Selecting at most 1 of 2 branches
                                        IF: Branch not selected
                                        ELSE: Entity 31 sent to 17$
   10 17$               TRACE
   11 116$              ASSIGN
                                        CRITICAL RATIO RULE Z VALUE set to 10.0317
                                        PROCESSING TIME set to 3.77704
   12 1$                TRACE
   13 117$              BRANCH
                                        Selecting at most 1 of 7 branches
                                        IF: Branch not selected
                                        IF: Branch not selected
                                        IF: Branch not selected
                                        IF: Entity 31 sent to 8$
   95 8$                STATION
                                        Entity 31 entered station MACHINE 4
   96 616$              TRACE
   97 579$              DELAY
                                        Delayed by 0.0 until time 1808.93
   98 623$              TRACE
   99 540$              QUEUE
                                        QUEUETIME set to 1808.93
                                        Entity 31 sent to next block
  100 541$              SEIZE
                                        Entity 31 added to queue MACHINE 4_R_Q at rank 2
Time: 1814.58  Entity: 31
  101 650$              BRANCH
                                        Selecting at most 1 of 2 branches
                                        IF: Branch not selected
                                        IF: Entity 31 sent to 553$
  103 553$              TALLY
                                        Tally MACHINE 4_R_Q QUEUE TIME recorded 5.65019
  104 660$              DELAY
                                        Delayed by 0.0 until time 1814.58
  105                   TRACE
  106 542$              DELAY
                                        Delayed by 3.77704 until time 1818.36
Time: 1818.36  Entity: 31
  107 624$              TRACE
  108 543$              RELEASE
                                        MACHINE 4_R available increased by 1 to 1
                                        Entity 29 removed from queue MACHINE 4_R_Q
                                        Resource allocated to entity 29
                                        Seized 1 unit(s) of resource MACHINE 4_R
  109 607$              DELAY
                                        Delayed by 0.0 until time 1818.36
  110 630$              TRACE
                                        Entity transferred to block 10$
   30 10$               TRACE
```

```
31 239$           ASSIGN
                            NUMBER OF OPERATIONS COMPLETED set to 4.0
                            PREVIOUS MACHINE set to 4.0
32 29$            TRACE
33 240$           BRANCH
                            Selecting at most 1 of 2 branches
                            IF: Branch not selected
                            ELSE: Entity 31 sent to 28$
38 28$            TRACE
39 243$           BRANCH
                            Selecting at most 1 of 2 branches
                            IF: Branch not selected
                            ELSE: Entity 31 sent to TEST
57 test           TRACE
58 293$           TALLY
                            Tally TARDINESS recorded 7.42088
59 34$            TRACE
60 295$           TALLY
                            Tally MAD FROM DUE DATE recorded 7.42088
61 35$            TRACE
62 297$           ASSIGN
                            NUMBER OF TARDY JOBS set to 230.0
                            Entity transferred to block 25$
46 25$            TRACE
47 249$           TALLY
                            Tally FLOW TIME recorded 67.4209
48 24$            TRACE
49 251$           COUNT
                            Counter FINISHED JOBS incremented by 1 to 431
50 19$            TRACE
51 253$           ASSIGN
                            FRACTION OF TARDY JOBS set to 0.533643
                            SYSTEM INVENTORY set to 21.0
52 18$            STATION
                            Entity 31 entered station DEPART
53 284$           TRACE
54 254$           DELAY
                            Delayed by 0.0 until time 1818.36
55 291$           TRACE
56 283$           DISPOSE
                            Disposed entity 31
```

Appendix C – Regression Code and Model

The regression code was developed in SAS using the PROCREG command. Four separate regression models were developed for each of the performance measures. The following is the code.

```
data work.new; set work.i40;
f01     =       k       ;
f02     =       k*k     ;
f03     =       k*k*k   ;
f04     =       k*k*k*k ;
f05     =       pa      ;
f06     =       pa*k    ;
f07     =       pa*k*k  ;
f08     =       pa*k*k*k        ;
f09     =       pa*pa   ;
f10     =       pa*pa*k ;
f11     =       pa*pa*k*k       ;
f12     =       pa*pa*pa        ;
f13     =       pa*pa*pa*k      ;
f14     =       pa*pa*pa*pa     ;
f15     =       pa*pa*pa*pw     ;
f16     =       pa*pa*pa*u      ;
f17     =       pa*pa*pw        ;
f18     =       pa*pa*pw*k      ;
f19     =       pa*pa*pw*pw     ;
f20     =       pa*pa*pw*u      ;
f21     =       pa*pa*u ;
f22     =       pa*pa*u*k       ;
f23     =       pa*pa*u*u       ;
f24     =       pa*pw   ;
f25     =       pa*pw*k ;
f26     =       pa*pw*k*k       ;
f27     =       pa*pw*pw        ;
f28     =       pa*pw*pw*k      ;
f29     =       pa*pw*pw*pw     ;
f30     =       pa*pw*pw*u      ;
f31     =       pa*pw*u ;
f32     =       pa*pw*u*k       ;
f33     =       pa*pw*u*u       ;
f34     =       pa*u    ;
f35     =       pa*u*k  ;
f36     =       pa*u*k*k        ;
f37     =       pa*u*u  ;
f38     =       pa*u*u*k        ;
f39     =       pa*u*u*u        ;
f40     =       pw      ;
f41     =       pw*k    ;
f42     =       pw*k*k  ;
f43     =       pw*k*k*k        ;
f44     =       pw*pw   ;
f45     =       pw*pw*k ;
f46     =       pw*pw*k*k       ;
f47     =       pw*pw*pw        ;
f48     =       pw*pw*pw*k      ;
f49     =       pw*pw*pw*pw     ;
f50     =       pw*pw*pw*u      ;
f51     =       pw*pw*u ;
f52     =       pw*pw*u*k       ;
```

```
f53     =       pw*pw*u*u       ;
f54     =       pw*u    ;
f55     =       pw*u*k  ;
f56     =       pw*u*k*k        ;
f57     =       pw*u*u  ;
f58     =       pw*u*u*k        ;
f59     =       pw*u*u*u        ;
f60     =       u       ;
f61     =       u*k     ;
f62     =       u*k*k   ;
f63     =       u*k*k*k ;
f64     =       u*u     ;
f65     =       u*u*k   ;
f66     =       u*u*k*k ;
f67     =       u*u*u   ;
f68     =       u*u*u*k ;
f69     =       u*u*u*u ;
run;
proc reg data=work.new;
   model MAD  = f01 f02 f03 f04 f05 f06 f07 f08 f09 f10 f11 f12 f13 f14 f15 f16 f17 f18
f19 f20 f21 f22 f23 f24 f25 f26 f27 f2 f29 f30 f31 f32 f33 f34 f35 f36 f37 f38 f39 f40
f41 f42 f43 f44 f45 f46 f47 f48 f49 f50 f51 f52 f53 f54 f55 f56 f57 f58 f59 f60 f61 f62
f63 f64 f65 f66 f67 f68 f69;
   output out=result1 p=pred r=resid;
run;
proc reg data=work.new;
   model SI  = f01 f02 f03 f04 f05 f06 f07 f08 f09 f10 f11 f12 f13 f14 f15 f16 f17 f18
f19 f20 f21 f22 f23 f24 f25 f26 f27 f2 f29 f30 f31 f32 f33 f34 f35 f36 f37 f38 f39 f40
f41 f42 f43 f44 f45 f46 f47 f48 f49 f50 f51 f52 f53 f54 f55 f56 f57 f58 f59 f60 f61 f62
f63 f64 f65 f66 f67 f68 f69;
   output out=result2 p=pred r=resid;
run;
proc reg data=work.new;
   model FT  = f01 f02 f03 f04 f05 f06 f07 f08 f09 f10 f11 f12 f13 f14 f15 f16 f17 f18
f19 f20 f21 f22 f23 f24 f25 f26 f27 f2 f29 f30 f31 f32 f33 f34 f35 f36 f37 f38 f39 f40
f41 f42 f43 f44 f45 f46 f47 f48 f49 f50 f51 f52 f53 f54 f55 f56 f57 f58 f59 f60 f61 f62
f63 f64 f65 f66 f67 f68 f69;
   output out=result3 p=pred r=resid;
run;
proc reg data=work.new;
   model f  = f01 f02 f03 f04 f05 f06 f07 f08 f09 f10 f11 f12 f13 f14 f15 f16 f17 f18
f19 f20 f21 f22 f23 f24 f25 f26 f27 f2 f29 f30 f31 f32 f33 f34 f35 f36 f37 f38 f39 f40
f41 f42 f43 f44 f45 f46 f47 f48 f49 f50 f51 f52 f53 f54 f55 f56 f57 f58 f59 f60 f61 f62
f63 f64 f65 f66 f67 f68 f69;
   output out=result4 p=pred r=resid;
run;
```

The following is the regression model generated by the code above

```
                              The REG Procedure
                                Model: MODEL1
                          Dependent Variable: MAD

                             Analysis of Variance

                                      Sum of           Mean
     Source                 DF       Squares         Square    F Value    Pr > F

     Model                  59     250.43070        4.24459      15.87    <.0001
     Error                 265      70.88161        0.26748
     Corrected Total       324     321.31231


              Root MSE              0.51718    R-Square     0.7794
              Dependent Mean        0.96077    Adj R-Sq     0.7303
              Coeff Var            53.83005
```

NOTE: Model is not full rank. Least-squares solutions for the parameters are not unique. Some
statistics will be misleading. A reported DF of 0 or B means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a linear combination of other variables as shown.

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	B	506.25221	202.86491	2.50	0.0132
f01	B	12.96639	13.18628	0.98	0.3263
f02	B	-3.12220	1.26766	-2.46	0.0144
f03	B	0.19499	0.09797	1.99	0.0476
f04	B	-0.00508	0.00362	-1.40	0.1616
f05	B	2.24395	4.93612	0.45	0.6498
f06	B	-0.10690	0.34887	-0.31	0.7595
f07	B	-0.03430	0.02766	-1.24	0.2160
f08	B	0.00086972	0.00117	0.75	0.4564
f09	B	-0.00037676	0.12043	-0.00	0.9975
f10	B	-0.00209	0.00821	-0.25	0.7995
f11	B	0.00028715	0.00045884	0.63	0.5320
f12	0	0	.	.	.
f13	0	0	.	.	.
f14	0	0	.	.	.
f15	B	-0.00614	0.01087	-0.56	0.5729
f16	B	-2.29584	0.86169	-2.66	0.0082
f17	B	0.21553	0.40678	0.53	0.5967
f18	B	0.00116	0.00115	1.01	0.3115
f19	B	-0.00077573	0.00190	-0.41	0.6837
f20	B	0.01429	0.02412	0.59	0.5541
f21	B	86.15680	32.32902	2.66	0.0082
f22	B	-0.00625	0.00814	-0.77	0.4435
f23	B	-0.03847	0.20652	-0.19	0.8524
f24	B	-2.14359	4.33996	-0.49	0.6218
f25	B	-0.01254	0.04766	-0.26	0.7927
f26	B	-0.00014070	0.00226	-0.06	0.9504
f27	B	0.04091	0.07063	0.58	0.5629
f28	B	-0.00113	0.00214	-0.53	0.5973
f29	0	0	.	.	.
f30	B	-0.01029	0.04527	-0.23	0.8204
f31	B	-0.56153	1.52003	-0.37	0.7121
f32	B	-0.00917	0.03919	-0.23	0.8153
f33	B	0.11066	0.92048	0.12	0.9044
f34	B	-956.51193	354.66748	-2.70	0.0074
f35	B	0.90110	0.67752	1.33	0.1847
f36	B	0.01677	0.02103	0.80	0.4258
f37	B	9.24784	23.35640	0.40	0.6925
f38	B	-0.59540	0.41094	-1.45	0.1486
f39	B	-2.39624	10.17084	-0.24	0.8139
f40	0	0	.	.	.
f41	B	-0.12427	0.93791	-0.13	0.8947
f42	B	0.15977	0.07486	2.13	0.0337
f43	B	-0.00551	0.00316	-1.74	0.0826
f44	B	-0.60914	1.22197	-0.50	0.6186
f45	B	0.00622	0.07712	0.08	0.9358
f46	B	-0.00214	0.00371	-0.58	0.5639
f47	0	0	.	.	.
f48	0	0	.	.	.
f49	0	0	.	.	.
f50	0	0	.	.	.
f51	B	0.56507	2.42720	0.23	0.8161
f52	B	0.03588	0.06567	0.55	0.5853
f53	B	-0.28312	1.45931	-0.19	0.8463
f54	B	28.68591	50.46476	0.57	0.5702
f55	B	-1.84233	1.77950	-1.04	0.3015
f56	B	-0.05955	0.05707	-1.04	0.2977
f57	B	-28.76265	64.33604	-0.45	0.6552
f58	B	1.51076	1.10733	1.36	0.1736
f59	B	9.41627	27.97431	0.34	0.7367
f60	0	0	.	.	.

f61	B	-4.58030	46.52632	-0.10	0.9217
f62	B	3.25814	2.01495	1.62	0.1071
f63	B	-0.09827	0.06015	-1.63	0.1035
f64	B	5896.75475	2150.49715	2.74	0.0065
f65	B	-12.45251	58.76844	-0.21	0.8324
f66	B	-0.66694	1.15550	-0.58	0.5643
f67	B	-5318.32105	1908.94811	-2.79	0.0057
f68	B	3.41778	25.72203	0.13	0.8944
f69	B	1795.28775	635.43502	2.83	0.0051

The REG Procedure
Model: MODEL1
Dependent Variable: SI

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	59	37.91911	0.64270	2.07	<.0001
Error	265	82.23166	0.31031		
Corrected Total	324	120.15077			

Root MSE	0.55705	R-Square	0.3156
Dependent Mean	-0.22769	Adj R-Sq	0.1632
Coeff Var	-244.65170		

NOTE: Model is not full rank. Least-squares solutions for the parameters are not unique. Some statistics will be misleading. A reported DF of 0 or B means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a linear combination of other variables as shown.

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	B	19.98174	218.50415	0.09	0.9272
f01	B	3.29651	14.20284	0.23	0.8166
f02	B	-0.15827	1.36538	-0.12	0.9078
f03	B	-0.00215	0.10552	-0.02	0.9837
f04	B	-0.00225	0.00390	-0.58	0.5644
f05	B	-11.90379	5.31665	-2.24	0.0260
f06	B	-0.18398	0.37577	-0.49	0.6248
f07	B	0.01317	0.02979	0.44	0.6587
f08	B	-0.00049636	0.00126	-0.40	0.6930
f09	B	0.32147	0.12972	2.48	0.0138
f10	B	-0.00502	0.00884	-0.57	0.5708
f11	B	0.00036996	0.00049422	0.75	0.4548
f12	0	0	.	.	.
f13	0	0	.	.	.
f14	0	0	.	.	.
f15	B	0.01632	0.01171	1.39	0.1646
f16	B	0.01326	0.92812	0.01	0.9886
f17	B	-0.58793	0.43814	-1.34	0.1808
f18	B	-0.00003091	0.00124	-0.02	0.9801
f19	B	-0.00091962	0.00205	-0.45	0.6539
f20	B	-0.02464	0.02598	-0.95	0.3438
f21	B	-1.34912	34.82132	-0.04	0.9691
f22	B	0.00054013	0.00877	0.06	0.9509
f23	B	0.56865	0.22244	2.56	0.0111
f24	B	5.01969	4.67454	1.07	0.2839
f25	B	-0.00442	0.05133	-0.09	0.9315
f26	B	-0.00301	0.00243	-1.24	0.2172
f27	B	-0.02016	0.07607	-0.26	0.7912
f28	B	-0.00026106	0.00231	-0.11	0.9100
f29	0	0	.	.	.
f30	B	0.06166	0.04876	1.26	0.2071
f31	B	3.75952	1.63722	2.30	0.0224
f32	B	0.05996	0.04221	1.42	0.1566

f33	B	-2.40046	0.99144	-2.42	0.0161
f34	B	47.97015	382.00946	0.13	0.9002
f35	B	0.47652	0.72975	0.65	0.5143
f36	B	-0.00746	0.02265	-0.33	0.7422
f37	B	-48.96879	25.15699	-1.95	0.0526
f38	B	-0.36694	0.44262	-0.83	0.4078
f39	B	18.57616	10.95493	1.70	0.0911
f40	0	0	.	.	.
f41	B	-0.20268	1.01021	-0.20	0.8411
f42	B	-0.01471	0.08063	-0.18	0.8554
f43	B	0.00198	0.00340	0.58	0.5606
f44	B	1.75888	1.31618	1.34	0.1826
f45	B	0.04452	0.08306	0.54	0.5924
f46	B	0.00347	0.00399	0.87	0.3858
f47	0	0	.	.	.
f48	0	0	.	.	.
f49	0	0	.	.	.
f50	0	0	.	.	.
f51	B	-4.34574	2.61432	-1.66	0.0976
f52	B	-0.11327	0.07074	-1.60	0.1105
f53	B	2.58302	1.57181	1.64	0.1015
f54	B	-61.46528	54.35518	-1.13	0.2592
f55	B	0.21958	1.91669	0.11	0.9089
f56	B	0.00232	0.06147	0.04	0.9699
f57	B	62.61137	69.29583	0.90	0.3671
f58	B	-0.13403	1.19270	-0.11	0.9106
f59	B	-19.72957	30.13090	-0.65	0.5132
f60	0	0	.	.	.
f61	B	-2.20202	50.11312	-0.04	0.9650
f62	B	0.63805	2.17028	0.29	0.7690
f63	B	0.06404	0.06479	0.99	0.3238
f64	B	-330.62675	2316.28303	-0.14	0.8866
f65	B	-13.38516	63.29901	-0.21	0.8327
f66	B	-1.14608	1.24458	-0.92	0.3580
f67	B	588.94315	2056.11252	0.29	0.7748
f68	B	16.11785	27.70499	0.58	0.5612
f69	B	-306.61411	684.42191	-0.45	0.6545

The REG Procedure
Model: MODEL1
Dependent Variable: FT

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	59	32.64596	0.55332	2.84	<.0001
Error	265	51.54635	0.19451		
Corrected Total	324	84.19231			

Root MSE	0.44104	R-Square	0.3878
Dependent Mean	-0.26923	Adj R-Sq	0.2514
Coeff Var	-163.81412		

NOTE: Model is not full rank. Least-squares solutions for the parameters are not unique. Some statistics will be misleading. A reported DF of 0 or B means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a linear combination of other variables as shown.

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	B	-32.75104	172.99722	-0.19	0.8500
f01	B	-5.76783	11.24487	-0.51	0.6084
f02	B	1.10484	1.08102	1.02	0.3077
f03	B	0.09826	0.08354	1.18	0.2406
f04	B	-0.00249	0.00309	-0.81	0.4203

f05	B	-2.43940	4.20938	-0.58	0.5627
f06	B	-0.14750	0.29751	-0.50	0.6204
f07	B	0.01134	0.02359	0.48	0.6310
f08	B	-0.00087461	0.00099435	-0.88	0.3799
f09	B	-0.02540	0.10270	-0.25	0.8049
f10	B	-0.00231	0.00700	-0.33	0.7419
f11	B	-0.00009536	0.00039129	-0.24	0.8076
f12	0	0	.	.	.
f13	0	0	.	.	.
f14	0	0	.	.	.
f15	B	0.00648	0.00927	0.70	0.4852
f16	B	0.19759	0.73482	0.27	0.7882
f17	B	-0.23331	0.34689	-0.67	0.5018
f18	B	0.00025469	0.00097949	0.26	0.7950
f19	B	-0.00009969	0.00162	-0.06	0.9510
f20	B	-0.01136	0.02057	-0.55	0.5812
f21	B	-7.37315	27.56923	-0.27	0.7893
f22	B	0.00345	0.00694	0.50	0.6198
f23	B	-0.00994	0.17611	-0.06	0.9550
f24	B	2.13853	3.70099	0.58	0.5639
f25	B	-0.03712	0.04064	-0.91	0.3619
f26	B	0.00001406	0.00193	0.01	0.9942
f27	B	-0.03039	0.06023	-0.50	0.6143
f28	B	-0.00010339	0.00183	-0.06	0.9549
f29	0	0	.	.	.
f30	B	0.04139	0.03860	1.07	0.2846
f31	B	1.52834	1.29624	1.18	0.2394
f32	B	0.04044	0.03342	1.21	0.2273
f33	B	-1.11390	0.78495	-1.42	0.1571
f34	B	96.44187	302.44997	0.32	0.7501
f35	B	0.45027	0.57777	0.78	0.4365
f36	B	0.01143	0.01793	0.64	0.5244
f37	B	-26.31382	19.91765	-1.32	0.1876
f38	B	-0.53599	0.35043	-1.53	0.1273
f39	B	14.87500	8.67340	1.72	0.0875
f40	0	0	.	.	.
f41	B	1.12907	0.79982	1.41	0.1592
f42	B	-0.09058	0.06384	-1.42	0.1571
f43	B	0.00215	0.00270	0.80	0.4250
f44	B	1.03994	1.04206	1.00	0.3192
f45	B	0.03118	0.06576	0.47	0.6358
f46	B	0.00145	0.00316	0.46	0.6464
f47	0	0	.	.	.
f48	0	0	.	.	.
f49	0	0	.	.	.
f50	0	0	.	.	.
f51	B	-2.14594	2.06985	-1.04	0.3008
f52	B	-0.05972	0.05601	-1.07	0.2873
f53	B	1.11518	1.24446	0.90	0.3710
f54	B	-41.67565	43.03486	-0.97	0.3337
f55	B	-1.17396	1.51751	-0.77	0.4399
f56	B	0.04171	0.04867	0.86	0.3923
f57	B	56.70533	54.86388	1.03	0.3023
f58	B	0.39735	0.94430	0.42	0.6742
f59	B	-22.96856	23.85567	-0.96	0.3365
f60	0	0	.	.	.
f61	B	3.38167	39.67627	0.09	0.9321
f62	B	-4.68571	1.71829	-2.73	0.0068
f63	B	-0.04338	0.05129	-0.85	0.3985
f64	B	-598.20065	1833.88058	-0.33	0.7445
f65	B	27.07930	50.11600	0.54	0.5894
f66	B	3.20747	0.98538	3.26	0.0013
f67	B	592.05105	1627.89468	0.36	0.7164
f68	B	-23.56779	21.93499	-1.07	0.2836
f69	B	-215.16249	541.88026	-0.40	0.6916

The REG Procedure
Model: MODEL1
Dependent Variable: f

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	59	36.67750	0.62165	12.37	<.0001
Error	265	13.32173	0.05027		
Corrected Total	324	49.99923			

Root MSE	0.22421	R-Square	0.7336
Dependent Mean	-0.07846	Adj R-Sq	0.6742
Coeff Var	-285.75939		

NOTE: Model is not full rank. Least-squares solutions for the parameters are not unique. Some statistics will be misleading. A reported DF of 0 or B means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a linear combination of other variables as shown.

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	B	-67.29895	87.94688	-0.77	0.4448
f01	B	-3.35824	5.71658	-0.59	0.5574
f02	B	-0.62736	0.54956	-1.14	0.2547
f03	B	0.09297	0.04247	2.19	0.0295
f04	B	-0.00347	0.00157	-2.21	0.0281
f05	B	-4.70276	2.13993	-2.20	0.0288
f06	B	-0.02907	0.15124	-0.19	0.8477
f07	B	-0.01540	0.01199	-1.28	0.2002
f08	B	0.00062622	0.00050550	1.24	0.2165
f09	B	0.10186	0.05221	1.95	0.0521
f10	B	0.00300	0.00356	0.84	0.4003
f11	B	-0.00018541	0.00019892	-0.93	0.3521
f12	0	0	.	.	.
f13	0	0	.	.	.
f14	0	0	.	.	.
f15	B	0.00574	0.00471	1.22	0.2241
f16	B	0.35340	0.37356	0.95	0.3450
f17	B	-0.21569	0.17635	-1.22	0.2224
f18	B	-0.00005574	0.00049794	-0.11	0.9110
f19	B	-0.00077144	0.00082459	-0.94	0.3504
f20	B	0.00468	0.01046	0.45	0.6551
f21	B	-13.54748	14.01542	-0.97	0.3346
f22	B	-0.00051071	0.00353	-0.14	0.8851
f23	B	0.19018	0.08953	2.12	0.0346
f24	B	1.87481	1.88148	1.00	0.3199
f25	B	-0.00164	0.02066	-0.08	0.9368
f26	B	0.00067648	0.00097941	0.69	0.4904
f27	B	0.02580	0.03062	0.84	0.4002
f28	B	-0.00033653	0.00092824	-0.36	0.7172
f29	0	0	.	.	.
f30	B	-0.00178	0.01962	-0.09	0.9277
f31	B	1.19875	0.65897	1.82	0.0700
f32	B	-0.00371	0.01699	-0.22	0.8274
f33	B	-0.84342	0.39905	-2.11	0.0355
f34	B	162.97434	153.75698	1.06	0.2901
f35	B	0.16865	0.29372	0.57	0.5663
f36	B	0.01057	0.00912	1.16	0.2472
f37	B	-20.14391	10.12557	-1.99	0.0477
f38	B	-0.16071	0.17815	-0.90	0.3678
f39	B	7.81589	4.40931	1.77	0.0774
f40	0	0	.	.	.

f41	B	0.39068	0.40661	0.96	0.3375
f42	B	0.01687	0.03245	0.52	0.6036
f43	B	-0.00147	0.00137	-1.07	0.2841
f44	B	0.04426	0.52976	0.08	0.9335
f45	B	-0.00050703	0.03343	-0.02	0.9879
f46	B	-0.00138	0.00161	-0.86	0.3902
f47	0	0	.	.	.
f48	0	0	.	.	.
f49	0	0	.	.	.
f50	0	0	.	.	.
f51	B	-0.54159	1.05225	-0.51	0.6072
f52	B	0.02718	0.02847	0.95	0.3407
f53	B	0.25811	0.63265	0.41	0.6836
f54	B	-21.88552	21.87770	-1.00	0.3181
f55	B	-1.19217	0.77146	-1.55	0.1235
f56	B	0.01010	0.02474	0.41	0.6834
f57	B	22.25920	27.89124	0.80	0.4255
f58	B	0.62318	0.48005	1.30	0.1954
f59	B	-6.03367	12.12754	-0.50	0.6192
f60	0	0	.	.	.
f61	B	18.62496	20.17029	0.92	0.3566
f62	B	-0.09718	0.87353	-0.11	0.9115
f63	B	-0.02447	0.02608	-0.94	0.3489
f64	B	-1054.12306	932.29286	-1.13	0.2592
f65	B	-21.40445	25.47755	-0.84	0.4016
f66	B	0.23046	0.50094	0.46	0.6459
f67	B	1067.60732	827.57547	1.29	0.1982
f68	B	7.58121	11.15113	0.68	0.4972
f69	B	-390.67133	275.47655	-1.42	0.1573

Appendix D – Neural Network MatLab Code and Results

Backpropagation neural networks can be designed in MatLab using many different algorithms, but because of the generalization importance to this research, an automated regularization process was used, the "trainbr" algorithm. The "trainbr" algorithm was built on the Bayesian framework of MacKay (1992). Combining this process with Levenberg-Marquardt training produced the best results compared with other training algorithms. In order to improve the generalization of the estimation method further, a technique called "early stopping" was implemented. A full discussion of early stopping can be found in neural network toolbox manual.

After investigating several neural networks architectures it was found that BPNN using 2 hidden layers with the use of the "tansig" transfer function produced more accurate estimates than other architectures. As for the output layer, the "purelin" transfer function produced better estimates. A full description of both transfer functions can be found in neural network toolbox manual.

The following is the MatLab code for the approximation process using BPNN. Three data sets were generated from the simulation output: training set, validation set, and testing set which used 50%, 25%, and 25% of the simulation data respectively.

```
function bpnn(p,t,e,a,b,c)
% usage:
%     bpnn(p,t,e,a,n1,n2)
% p: nn input
% t: nn targets
% e: early stoping (0,1) ===> (no, yes)
% a: (1,2,3,4) ===>(mad, si, ft, f)
% n1: number of neurons in hidden layer 1
% n2: number of neurons in hidden layer 2

t=t(a,:);
```

```
[pn,meanp,stdp,tn,meant,stdt]=prestd(p,t);

q=819;
if e==1
    iitr=[1:4:q 3:4:q];iival=4:4:q;iitst=2:4:q;
    train_P=pn(:,iitr);valid.P=pn(:,iival);test.P=pn(:,iitst);
    train_T=tn(:,iitr);valid.T=tn(:,iival);test.T=tn(:,iitst);
else
    iitr=1:2:q;
    train_P=pn(:,iitr);train_T=tn(:,iitr);
end

%creat the neural network
net = newff(minmax(train_P),[b c 1],{'tansig' 'tansig' 'purelin'},'trainbr');
net.trainparam.show    = 25;
net.trainParam.epochs = 200;
net.trainParam.goal    = 1e-5;

if e==1
    [net,tr] = train(net,train_P,train_T,[],[],valid,test);
    print(gcf,[num2str(a) 'plot1'],'-djpeg99','-r0');
    figure(1);
    plot(tr.epoch,tr.perf,tr.epoch,tr.vperf,tr.epoch,tr.tperf);grid;
    legend('Training','Validation','Test',-1);
    ylabel('Square Error');xlabel('Epoch');
    print(gcf,[num2str(a) 'plot2'],'-djpeg99','-r0');
else
    [net,tr] = train(net,train_P,train_T);
    print(gcf,[num2str(a) 'plot1'],'-djpeg99','-r0');
    figure(1);
    plot(tr.epoch,tr.perf);grid;
    legend('Training',-1);
    ylabel('Square Error');xlabel('Epoch');
    print(gcf,[num2str(a) 'plot2'],'-djpeg99','-r0');
end

sim_tn = sim(net,pn);
t_r = poststd(sim_tn,meant,stdt);

b=[-1:.25:2.5 0;histc((t-t_r)',-1.125:.25:2.625)';histc((t-t_r)',-
1.125:.25:2.625)'/8.19]'
b(4,3)+b(5,3)+b(6,3)
figure(2);hist((t-t_r)',-1:.25:2.5);
print(gcf,[num2str(a) 'plot3'],'-djpeg99','-r0');
figure(3);test_results(t,t_r);grid;
print(gcf,[num2str(a) 'plot4'],'-djpeg99','-r0');

fhd=fopen([num2str(a) 'pred.txt'],'w');
for i=1:q
    fprintf(fhd,'%f %f %f %f %f \n',p(:,i),t_r(i));
end
fclose(fhd);

return

function [r_sqrd] = test_results(t,sim_t)
[a,b]   = size(t);
t_bar   = sum(t)/b;
SSE     = sum((t-sim_t).^2);
SSTO    = sum((t-t_bar).^2);
r_sqrd= 1-(SSE/SSTO);

[d e f]=postreg(t,sim_t);

fprintf(1,'r squared for the whole set is %f\n',r_sqrd);
fprintf(1,'slope=%f, intercept=%f, r=%f',d,e,f);
return
```

The above code generates four graphs: training curves, square error graph, error histogram, and linear regression fitness test. Training curves and error histogram are useful in selecting the best neural network architectures. The square error graph and linear regression fitness tests give better indication of the accuracy for the architecture. The later two graphs are presented for each performance measure in Figure 12 through Figure 19.

Square error graphs present the error as a function of the training time. It is apparent that the training error is almost twice the validation and testing errors, this is due to the fact that the training set is twice the size of the validation and training sets. This graph is used for the early stopping technique; training is stopped once the validation error increases for a specified number of iterations.

The linear regression fitness test is useful to investigate the network response. Regression analysis between the network response and the corresponding targets is performed. The goal is to have a perfect fit that will have a slope of one and an intercept of zero and most of the values lie on the regression line. "A" is the predicted values, "T" is the target value.

A complete electronic copy of the programs, designs, and network weights can be obtained by contacting the Industrial Engineering & Management Department, Oklahoma State University, Stillwater, Ok.

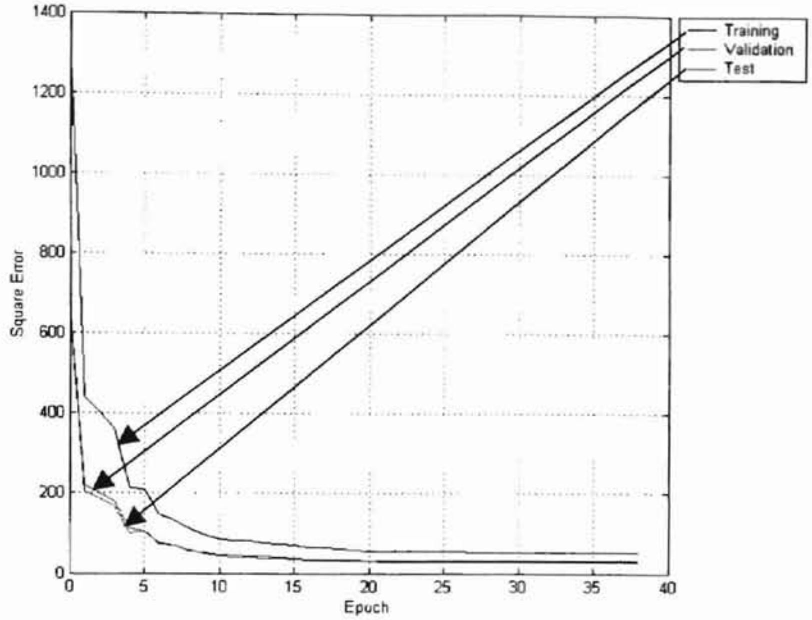

Figure 12 - Square error graph for training session for *MAD*

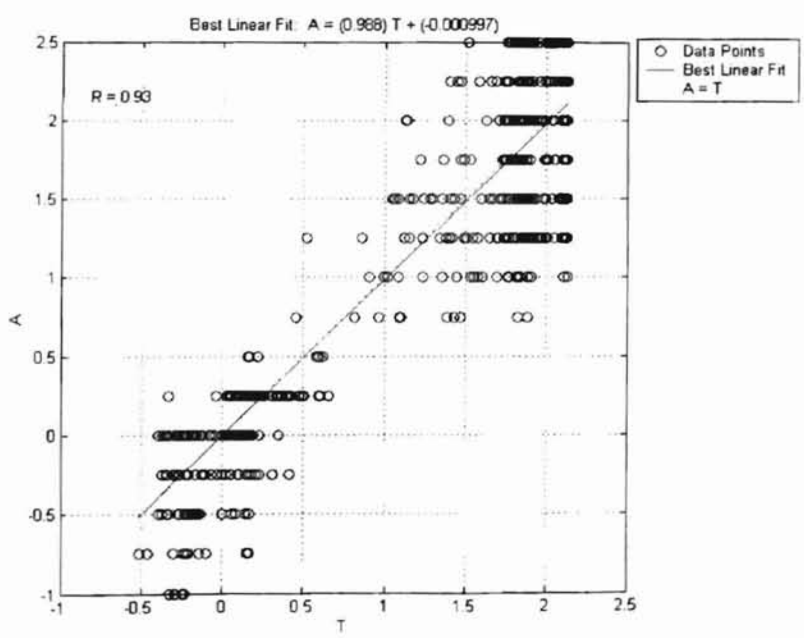

Figure 13 - Linear regression fitness tests for *MAD*

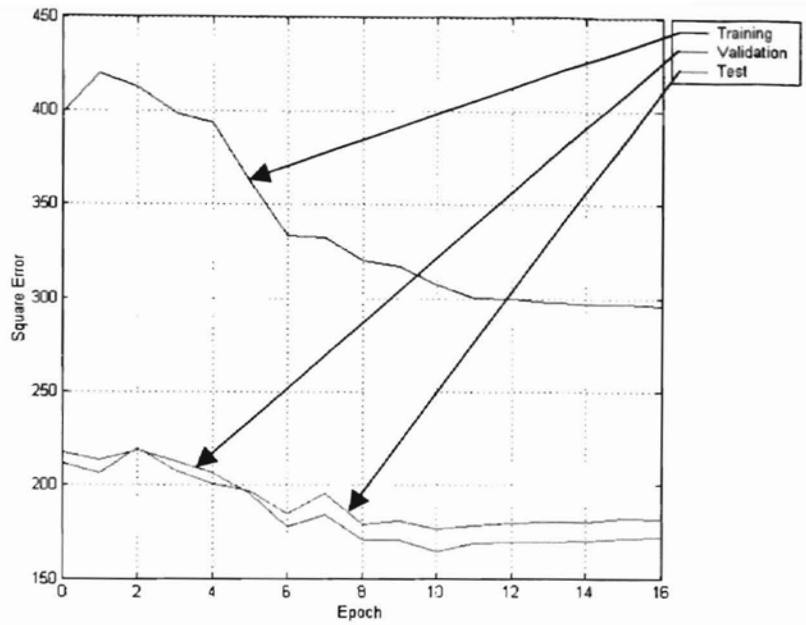

Figure 14 - Square error graph for training session for *S1*

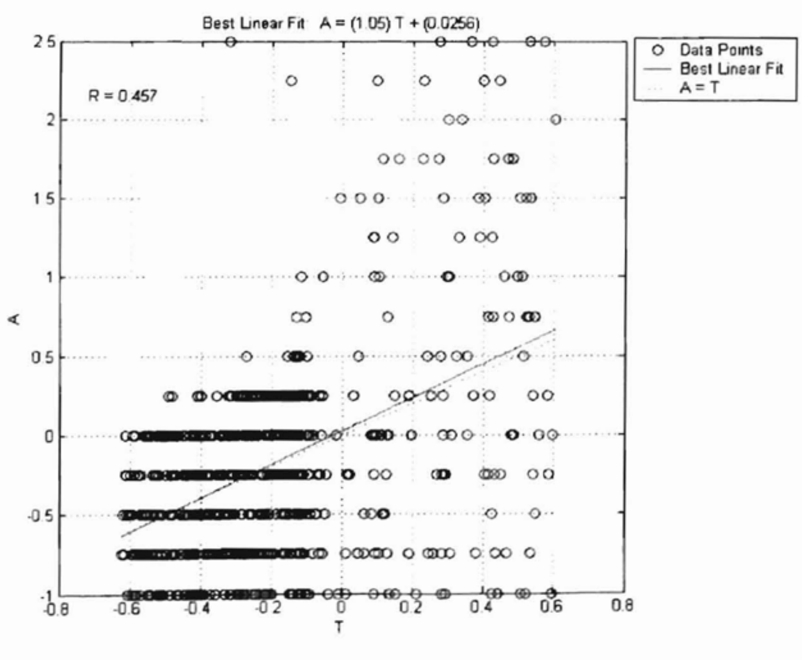

Figure 15 - Linear regression fitness tests for *S1*

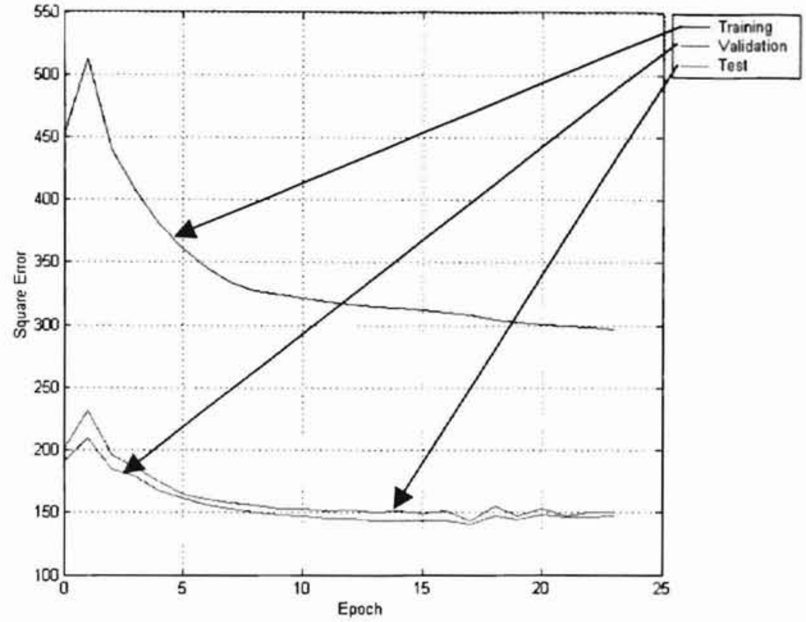

Figure 16 - Square error graph for training session for *FT*

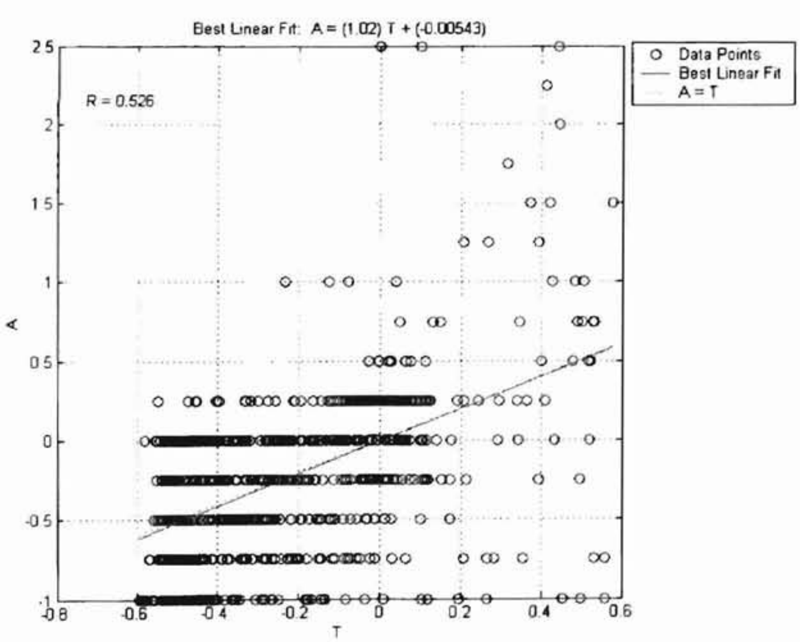

Figure 17 - Linear regression fitness tests for *FT*

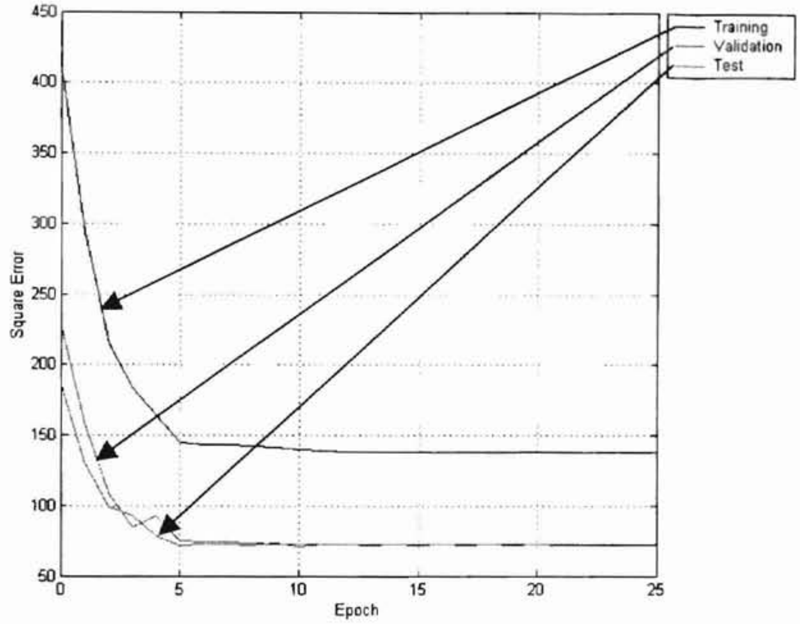

Figure 18 - Square error graph for training session for *f*

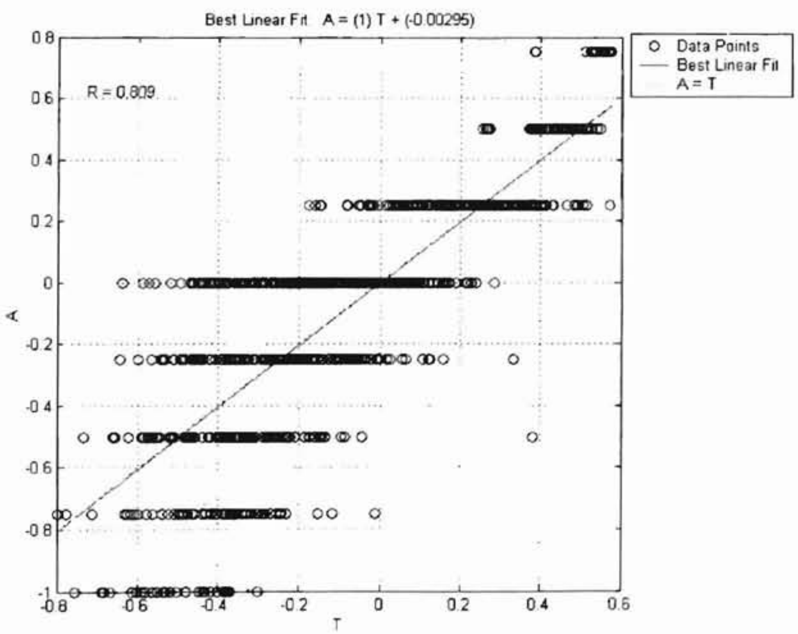

Figure 19 - Linear regression fitness tests for *f*

Appendix E – CRz, CR, and SPT Comparison Results

In order to perform comparisons between CR_z and both CR and SPT, the simulation model had to be modified to include adjustments in prioritizing rules. The 16 extreme cases were fed into the simulation model and the results were recorded in two separate runs; one for each sequencing rule.

Table 10 shows the full result of the neural networks and the two simulation runs. In order to avoid the performance measure's magnitude difference from one case to another improvement percentages were normalized using Equation 7.

$$Normalized\_Error = \frac{Performance\_measure\_(CR) - Performance\_measure\_(CRz)}{Performance\_measure\_(CRz)}$$

Equation 7

Table 11 shows the normalized improvement percentages due to the change of sequencing rules from CR / SPT to CR_z. It should be noted that changing the due date allowance factor (k) directly affect both SI and FT, this is due to the fact that early jobs are delayed until their due date which have an inflating effect on both SI and FT.

	Parameters				Performance measures using CRz (BPNN predicted z values)				Performance measures using CR				Performance measures using SPT			
Num	Pa	Pw	K	U%	*MAD*	*SI*	*FT*	*f*	*MAD*	*SI*	*FT*	*f*	*MAD*	*SI*	*FT*	*f*
1	5	0.5	3	60	33.6	12.7	75.5	0.025	33.8	12.9	76.7	0.067	34.4	12.7	75.8	0.038
2	5	0.5	3	90	61.9	33.0	130.9	0.636	73.3	35.5	141.5	0.651	69.6	33.4	132.9	0.467
3	5	0.5	9	60	180.0	37.6	224.8	0.000	180.4	37.7	225.0	0.000	182.9	37.7	225.0	0.000
4	5	0.5	9	90	132.5	60.7	241.1	0.084	153.9	67.9	269.9	0.174	145.1	62.0	246.8	0.101
5	5	2	3	60	33.3	12.7	75.5	0.027	33.5	12.9	76.7	0.070	34.2	12.7	75.9	0.041
6	5	2	3	90	60.8	32.4	128.9	0.650	76.9	36.8	145.8	0.673	73.4	34.6	137.1	0.477
7	5	2	9	60	179.6	37.6	224.7	0.000	180.0	37.8	224.9	0.000	182.4	37.8	224.9	0.000
8	5	2	9	90	131.8	60.9	241.5	0.085	156.3	69.6	275.0	0.183	145.9	63.6	251.2	0.113
9	20	2	3	60	133.9	12.7	301.9	0.024	134.5	12.9	306.8	0.066	137.4	12.8	303.6	0.037
10	20	2	3	90	246.6	32.8	521.1	0.604	306.8	36.6	580.9	0.625	293.6	34.4	547.1	0.443
11	20	2	9	60	719.3	37.6	899.1	0.000	722.6	37.8	901.0	0.000	732.3	37.8	900.9	0.000
12	20	2	9	90	528.6	60.7	963.4	0.077	609.8	67.6	1072.4	0.161	578.9	62.2	987.4	0.096
13	20	8	3	60	132.9	12.7	301.8	0.028	134.2	12.9	306.6	0.066	136.9	12.7	303.3	0.037
14	20	8	3	90	246.4	32.9	522.9	0.625	329.2	38.2	605.4	0.643	306.1	35.4	560.9	0.450
15	20	8	9	60	718.5	37.6	899.3	0.000	720.6	37.8	900.8	0.001	730.9	37.8	900.7	0.000
16	20	8	9	90	525.2	59.8	953.2	0.074	612.2	68.3	1083.1	0.172	574.8	62.2	986.9	0.099

Table 10 - The full list of parameters of the compared cases and the corresponding performance measures using CRz, CR, and SPT sequencing rules

VITA γ

Sinan Salman

Candidate for the Degree of

Master of Science

Thesis: JOB SHOP SCHEDULING: A PREDICTIVE NEURAL NETWORK MODELING SYSTEM FOR A QUANTIFIED SEQUENCING RULE

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Amman, Jordan, On April 25, 1977, the son of Mahmoud Salman and Zain Shawwash

Education: Received Bachelor of Science degree in Mechanical Engineering from Jordan University of Science and Technology, Irbid, Jordan; completed the requirements for the Master of Science degree in Industrial Engineering and Management at Oklahoma State University in April 2001.

Experience: Trainee Engineer at RUM metal industries, Sahab, Jordan, from June 1998 to September 1998; Teaching Assistant, Fortran Lab, Oklahoma State University, from August 2000 to December 2000; Graduate Research Assistant, Technology Forecasting Group, Advanced Demand Forecasting Group, Oklahoma State University, from January 2000 to May 2001.

Honors: Alpha Pi Mu Industrial Engineering Honor Society
Phi Kappa Phi Honor Society

Affiliations: Institute For Operations Research and Management Science (INFORMS)