Cost-Based Shop Control Using Artificial Neural Networks

by

Lars Wiegmann

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Management Science

APPROVED:

Loren P. Rees

Loren P. Rees

Philip Y. Huang

Philip Y. Huang

Lance A. Matheson

Lance A. Matheson

Jerry R. Rakes

Jerry R. Rakes

Robert T. Sumichrast

May 7, 1992

Blacksburg, Virginia

COST-BASED SHOP CONTROL USING ARTIFICIAL NEURAL NETWORKS

by

Lars Wiegmann

Loren P. Rees, Chairman

Management Science

(ABSTRACT)

The production control system of a shop consists of three stages: due-date prediction, order release, and job dispatching. The literature has dealt thoroughly with the third stage, but there is a paucity of study on either of the first two stages or on interaction between the stages. This dissertation focuses on the first stage of production control, due-date prediction, by examining methodologies for improved prediction that go beyond either practitioner or published approaches. In particular, artificial neural networks and regression nonlinear in its variables are considered. In addition, interactive effects with the third stage, shop-floor dispatching, are taken into consideration.

The dissertation conducts three basic studies. The first examines neural networks and regression nonlinear in its variables as alternatives to conventional due-date prediction. The second proposes a new cost-based criterion and prediction methodology that explicitly includes costs of earliness and tardiness directly in the forecast; these costs may differ in form and/or degree from each other. And third, the benefit of tying together the first and third stages of production control is explored. The studies are conducted by statistically analyzing data generated from simulated shops.

Results of the first study conclude that both neural networks and regression nonlinear in its variables are preferred significantly to approaches advanced to date in the literature and in practice. Moreover, in the second study, it is found that the consequences of not using the cost-based criterion can be profound, particularly if a firm's cost function is asymmetric about the due date. Finally, it is discovered that the integrative, interactive methodology developed in the third study is significantly superior to the current non-integrative and non-interactive approaches. In particular, interactive neural network prediction is found to excel in the presence of asymmetric cost functions, whereas regression nonlinear in its variables is preferable under symmetric costs.

Acknowledgements

I would like to thank my chairman, Professor Loren Paul Rees, who has provided much more to me than guidance and assistance during my graduate studies. He has contributed with unsurmountable creativity, insight, and expertise to the planning and execution of this dissertation, constantly pressing for progress and striving for success. More importantly, however, his congenial personality and genuine character have been an example and inspiration to me, an experience I will benefit from throughout my future professional and personal life. In fact, in the three years we have worked together, Loren has truly become a special friend.

Furthermore, I would like to thank Dr. Philip Y. Huang, Dr. Lance A. Matheson, Dr. Terry R. Rakes, and Dr. Robert T. Sumichrast for also serving on my committee, and who all, along with others at Virginia Tech, have made this dissertation a very enjoyable learning experience.

Table of Contents

Chapter 1:

Chapter 2:

Chapter 4:

Chapter 5:

Chapter 6:

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Purpose and Justification

Shop control is receiving increasing attention from both researchers and practicing managers due to emphasis on Total Quality Management, Just-in-Time concepts, the Zero Inventory philosophy, etc. Cost control, in particular, has emerged as a necessary means for survival for many companies, with techniques for production control being an important component of this effort.

The order-promising, or due-date assignment decision is one facet of the shop-control problem. This facet is very important to the firm, as Wein (1991) [81], Cheng and Gupta (1989) [14], and others point out. Cheng and Gupta note:

> A review of the literature reveals that this aspect of the scheduling decision [determination of optimal due-date values] is of particular importance to both researchers and practicing managers. Research in this area obviously has not been carried out to its completeness since it is apparent that so many areas still remain untouched. The practicing managers are increasingly faced with difficult situations in which jobs have to be delivered on time, otherwise costs will be incurred. Minimizing inventories is the name of the game in today's extremely competitive business world. It has led to the concept of 'zero inventory' (ZI) which insists on carrying no inventory at all, whether it be raw material or supplies related inventory or finished goods inventory.

Increasingly, artificial intelligence (AI) techniques are finding their way into industry. One such technique is neural networks, a methodology originally based on the way the human brain is thought to operate. Successful industrial applications of neural networks have been in areas as diverse as pattern and speech recognition, image compression, visual systems, and signal processing. But almost all applications to date have been to non-managerial problems.

This research investigates the applicability of neural networks in the managerial decision area of due-date prediction. The purpose of this dissertation is to gain further understanding of due-date prediction in an integrated, shop control context, using cost-based measures. Attempts will be made to incorporate more advanced prediction techniques, which Cheng and Gupta infer are needed, to improve the order-promising function.

The basic methodology of the dissertation will be to use computer simulation to generate hypothetical shop data. These data will in turn be used to investigate hypotheses exploring best technique, effects of cost-based predictions, desirability of integrating the entire production control system, etc. Of course, building mathematical models of various predictors with and without cost-based due-date setting, is a necessary step before the hypotheses are investigated with appropriate statistical tests.

In order to define more precisely the objectives, scope, and limitations of this dissertation, more detailed discussion of the production control system as well as artificial neural networks is needed.

The Production Control System

As first suggested by Baker in 1984 [5] and reiterated by Ragatz and Mabert in 1988 [61], the production control system can be viewed as consisting of three sequential stages (see Figure 1-1): the order promising/master-scheduling stage, the order-release portion, and the shop-floor stage. In stage 1, order-promising/master-scheduling, jobs arrive from the customer and are assigned a due date. If due dates are specified by the production department, they are referred to as "internally-set due dates;" conversely, if due dates are set by an order-entry marketing department in agreement with the customer, they are called "externally-imposed due dates" (see, for example, [62]). This distinction emphasizes the control, or lack thereof, of the production department in its effort to meet its performance objectives. In the second stage, order release, work is sent -- possibly after some intentional delay -- to the shop floor. In the final stage the work itself is performed; during this stage the sequence of work at various constrained workcenters must be determined. This decision is referred to as "dispatching."

In this research the concentration is on the first stage of the production control system, namely order promising, and on the prediction of internally-set due dates in particular. Conventionally used techniques for due-date prediction, by researchers as well as practitioners, are almost exclusively regression-based. Ragatz and Mabert (1984) [62] published a comprehensive comparison of different due-date assignment rules, the best performers of which serve as a benchmark for this research. Some of the conventional regression-based rules considered include job characteristics only, while others include shop data as well. The sophistication of the regression-based rules considered in this research ranges from simple regression on only one independent

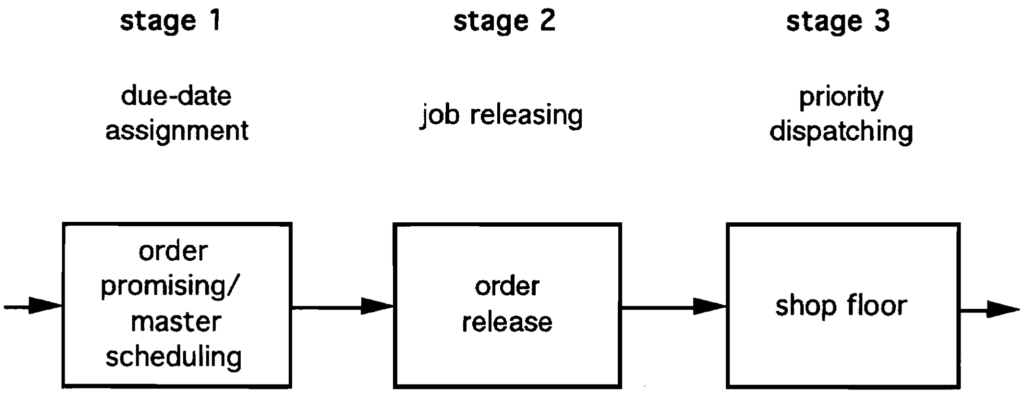


Figure 1-1: The three stages of the production control system

variable to a response mapping rule in which important independent variables have to be identified first before the various functional rule equations can be estimated. In addition, nonlinear independent variables are considered as well. As indicated, a primary focus of this dissertation is to compare neural networks as a new methodology for internally setting due dates to these conventionally used regression-based techniques.

In the second stage, order release, no intentional delay is assumed throughout this research. Some researchers have found at least qualitative advantages (see, for example, Ragatz and Mabert 1988 [61]) of holding jobs back from the shop floor in stage two, while others have raised serious doubts about order release altogether (see, for example, Baker 1984 [5]). Because of this debate, and because we had other purposes, the issue of order release was not studied.

Due to the complexity of shop floor control with different jobs having a variety of processing requirements and traveling between workcenters with different capabilities, it is not in general possible to specify an optimal control procedure for stage 3. As such, heuristic dispatching rules are used instead. For example, jobs arriving at workcenters may be sequenced according to a first-come, first-served (FCFS) queueing discipline, a shortest processing time (SPT) requirement, or with a procedure such as earliest-due-date (EDD) ranking. The first two examples just cited are illustrative of non-due-date-based rules in that the position a job takes in the queue does not depend on the due date of the job, but rather on some of job or shop characteristics. Alternatively, the last rule mentioned, EDD, does depend on a job's due date for queue discipline.

In this research the simpler case of non-due-date-based rules is studied in chapters 3 and 4, whereas a due-date-based rule is explored in chapter 5. By nature of the latter case, the order promising stage interacts with the dispatching stage. This interaction between stages is studied in chapter 5, thereby further setting it apart from other shop control and prediction research, which typically examines a single stage at a time.

Artificial Neural Networks

There are many different types of neural networks, including Hopfield, Brain-State-in-a-Box, Bidirectional Associative Memory, Boltzmann, Adaptive Resonance Theory, Hamming, and Spatiotemporal Networks ([31], [42], [53], [74], [78]). Probably the most common and readily available of the neural networks is the backpropagation network. This network has been used with success in many applications including forecasting [42], the area of interest here. Because the standard backpropagation network has achieved forecasting success in other domains, and because it is probably the most widely known neural network, it is used in this research. This choice is consonant with one of the ultimate objectives, to make advantageous artificial intelligence approaches usable for practitioners.

Backpropagation networks consist of an input layer, an output layer, and one or more "hidden" (middle) layers of processing elements. Processing elements (nodes) in each layer are generally connected to nodes in other layers. Each connection contains a weight, which is what is adjusted as neural networks learn. Backpropagation networks are characterized by supervised learning, whereby training data are used to "teach" the network. In particular, each presentation of training data will include a set of inputs and the corresponding desired set of outputs. If the actual output obtained from the network by presentation of the training inputs differs from the desired output, then the weights interconnecting the neurodes are adjusted in a fashion to bring the actual output closer in line to the desired output. A particular training data set often must be presented tens or hundreds of times to the network before the underlying relationship between inputs and desired outputs is learned.

For the application of neural networks in this dissertation training data are generated by means of simulation runs. These data would be available in the form of historical data in practical applications. The actual network output provides the predicted due-date, whereas the desired output is given as the actual completion time of the respective job as observed in the shop simulation. The difference between the actual flow time through the shop representing the known, desired output and the output obtained from the neural network defines an error that is used to modify the weights.

In general, the error function utilized in backpropagation networks depends on problem characteristics. In chapter 4 cost considerations are explicitly incorporated into the error function. That is, the error function is stipulated as the cost function of missing the due date: a tardiness function for positive values of error, and an earliness function for negative error. The error as modified by the cost function is then propagated back through the network (during the so-called "backward pass"), and the weights are adjusted to map inputs into the output more accurately the next time. In this way, due dates are forecast with the relevant cost functions in mind.

Objectives of the Research

As mentioned, Cheng and Gupta (1989) [14] pointed out that only relatively unsophisticated techniques have been used in setting due dates. The first, and primary, objective of this research is to see whether neural networks hold any promise for the area of due-date prediction, the first stage of shop control. In beginning the analysis and discussion of neural network applicability in this area, this research compares the performance of neural networks to regression-based prediction rules across a variety of shops and different data sample sizes. It should be noted that, because the literature is so sparse in the area of nonlinear models altogether, nonlinear variables are added to the regression techniques as another case for the comparisons. The initial goal is to gain some confidence in the applicability of neural networks in this environment.

A second objective of this research is to introduce cost measures into the prediction scheme of the due-date assignment techniques. This breaks with traditional assumptions of equal cost-of-earliness and cost-of-tardiness functions in the literature. For example, ordinary least square (OLS) regression inherently minimizes a quadratic cost function symmetric about the due-date, thereby assigning equal second-order costs to early completion and tardy behavior. Accordingly, the criteria mean absolute deviation (MAD) and standard deviation of lateness (SDL) implicitly assume symmetric linear and quadratic costs, respectively. Specifically evaluated in this dissertation are cost functions that differ in magnitude and/or shape on the two sides of the due date. The cost-based model, in fact, can be used with any differentiable earliness and tardiness cost functions.

Finally, the third objective of this research is to study the interaction between the two main stages of shop control, due-date prediction and shop-floor dispatching. Due to the complexity of the production control system, most research has concentrated on only one stage of production control at a time, while ignoring the interaction that is present when a due-date based dispatching rule is used in stage 3. As part of the whole framework of this dissertation, this interaction is studied with special focus on the performance of neural networks compared to conventional, regression-based techniques.

Scope and Limitations

As ambitious as the objectives stated above are, there are limitations to the research. In particular, there are three broad limitations.

First, the study is conducted on simulation data drawn from only a few shops. Hence results are not generalizable to all shops; rather, conclusions apply only to environments similar to those studied.

Second, this research makes no claim that the best neural network implementation has been used. In fact, no attempt has been made to go beyond "pure vanilla" backpropagation. This is because this is a feasibility study and also because the neural network technique was often found to be most preferred nonetheless.

The scope of this research is theoretical in its focus, and that leads to the third general limitation of our procedure. To date, no effort has been expended to implement results on real shops with real cost functions and real jobs. No difficulty is foreseen; nevertheless, implementation considerations have not been studied here.

Plan of Presentation

Chapter Description

One: is an introduction to the conceptual framework of the research. It outlines the production control system, provides some neural network basics and specifies objectives as well as scope and limitations of this dissertation.

Two: gives a review of the relevant literature. It is divided into the two areas of production control and neural networks which are merged in this research.

Three: justifies the applicability of neural networks for shop control and also suggests another non-linear model for study. A procedure of using neural networks for due-date prediction is outlined thoroughly. The performance of the neural network, a proposed nonlinear regression model, and six conventional regression-based procedures found in the literature are compared statistically on three hypothetical shops for varying sample sizes.

Four: suggests the use of cost functions as part of the due-date prediction scheme and statistically compares neural networks and classical approaches on a cost basis under several scenarios. It is explained why cost functions should and how they can be made an integral part of the due-date prediction procedure. Ordinary least squares regression, linear and quadratic programming, and neural networks are compared, and the monetary consequences of their use are specified for different cost cases.

Five: studies the effect of interaction between stages 1 and 3 of the production control system on the performance of various due-date prediction techniques. A procedure is outlined for both neural networks as well as regression-based due-date prediction techniques that accounts for the interaction between prediction and due-date dispatching. The efficiency of interactive versus non-interactive prediction is examined for five different prediction techniques over four different cost scenarios. Then, based on these results the cost performance of nonlinear and linear regression as well as neural networks and the best regression technique are statistically compared for the interactive due-date prediction case.

Six: presents a summary of results and contributions, and identifies some areas for future research.

Chapter 2

LITERATURE REVIEW

A primary purpose of this research is to investigate the potential of neural networks as applied to due-date prediction, the first stage of shop control. The review of related literature has been divided into two parts, reflecting the two major areas merged in this research. The first part deals with the relevant shop scheduling literature. It first addresses the production control system in general, and then the due-date decision in particular. Finally, the concept of cost performance criteria as developed in the literature is assessed. The second part reviews relevant research on neural networks by first looking at their historical background, and then at a few neural network models and some of their widely ranging applications.

Shop Scheduling Literature

The Production Control System

As first suggested by Baker in 1984 [5] and reiterated by Ragatz and Mabert in 1988 [61], the production control system can be viewed as consisting of three sequential stages: the order promising/master-scheduling stage, the order-release portion, and the shop-floor stage. In stage 1, order-promising/master-scheduling, jobs arrive from the customer and are assigned a due date. If due dates are specified by the production department, they are referred to as "internally-set due dates;" conversely, if due dates are set by an order-entry marketing department in agreement with the customer, they are called "externally-imposed due dates" (see, for example, [62]). This distinction emphasizes the control, or lack thereof, of the production department in its effort to meet its performance objectives. In the second stage, order release, work is sent -- possibly after some intentional delay -- to the shop floor. In the final stage the work itself is performed; during this stage the sequence of work at various constrained workcenters must be determined. This decision is referred to as "dispatching."

Because of the complexity of the production control system, most scheduling research has only examined one of the three stages at a time; that is, research has been performed regarding each of the three stages in isolation from the others. Most early work dealing with the due-date assignment decision, the first stage, examined only job characteristic information, while ignoring shop-status considerations. Some of this work was done by Conway [16], Eilon and Chowdhury [20], and Baker and Bertrand [6]. More recently, shop status conditions have been included as well. An excellent survey of due-date assignment research as of 1989 is presented by Cheng and Gupta [14]. Other relevant work includes that from Elvers [22], Kanet and Christy [41], Ragatz and Mabert [60], [61] and [62], Smith and Seidmann [76], and Weeks and Fryer [80]. As the first stage of the production control system is the focus of this dissertation, it will be returned to this topic in more detail momentarily.

With respect to stage two of the control system, several articles discuss order release as a concept to level the load in the shop (see, for example, Bechte [7], Fry [24], Fry and Smith [25], Kanet [39]). Reasons cited for the use of release control include reduced congestion as well as better shop floor visibility. Among others, Ragatz and Mabert [61] found that limiting the number of orders on the shop floor reduces the importance of dispatching. Alternatively, Kanet [39] concluded that order release cannot reduce the overall system flowtime, when the time the order has to wait for release to the shop floor is taken into consideration. He reasoned that by holding jobs back from the shop floor, the probability for some machines to become idle is increased, thereby reducing the realized capacity of the shop and increasing average flowtime. Kanet concluded that a cautious approach should be taken towards order release as customer delivery times can increase. Since this dissertation does not deal with order release per se (an immediate order-release mechanism is assumed throughout), several additional relevant papers are merely listed: Ackerman [1], Bertrand [8], Bobrowski and Park [10], Irastorza and Deanne [37], Melnyk and Ragatz [49], and Wight [84].

Finally, of all the stages, most research has been devoted to the third stage, shop floor control (dispatching). As there is no closed-form solution to the dispatching problem, the approach taken in the literature has been to study heuristic rules. For the purposes of this dissertation, it is convenient to group these rules into two main categories: due-date-based dispatching rules, and non-due-date-based rules. With due-date-based rules, jobs are sequenced at workcenters according to their due dates. For example, with earliest-due-date (EDD) dispatching, jobs line up at workcenters with the most-imminently-due jobs first and less pressing jobs farther back in the queue. Another due-date rule, MINSLK, orders jobs based on their slack time, which is the difference between time remaining until the due date and total processing time, with least slack jobs going first. Conversely, non-due-date-based rules determine placement of jobs in queues according to criteria other than due dates. For example, FIFO processes jobs on a first-come, first-served basis, which has little to do with the due date. Likewise, the shortest processing time dispatching rule (SPT) has a queue discipline set according to the jobs' required processing times at the workcenter, rather than basing the decision on due dates.

This dissertation, as explained, focuses on the first stage of shop control; the research merely assumes an SPT dispatching rule for two chapters, and utilizes the EDD rule for a third. Consequently, the dispatching literature is reviewed summarily, and thereby the myriad of dispatching articles is just listed without discussion: Baker [4], Blackstone et al. [9], Christy and Kanet [15], Conway [16], Day and Hottenstein [18], Elmaghraby [21], Graves [26], Gupta and Kyparisis [28], Jones [38], Kanet [40], Moore and Wilson [52], Panwalkar and Iskander [57], Scudder and Hoffman [71], Wein [81].

Although the majority of the production-control-system research has examined each of the three stages alone, a few writers have considered interaction among the stages. Elvers as early as 1973 [22] reported an important interaction between the determination of due dates and sequencing on the shop floor. Baker in 1984 [4] showed an interaction between the due-date assignment method and the use of milestones with the sequencing function. Cheng in 1988 [13] concluded that improved performance can be achieved by coordination and integration of the first and third stage.

To this point, an overview of the production-contol-system literature has been given. A review of research on each of the three stages taken alone has been provided, as has recognition of the few articles beginning to consider the control system as an integrated whole. Emphasis now shifts to a detailed review of two topics that will be dealt with specifically in later chapters of this dissertation, namely the due-date assignment decision, and cost performance criteria, as developed to date.

The Due-Date Assignment Decision

Assigning due dates is a difficult decision. As a job arrives at the shop, a due date is predicted indicating when the job is expected to complete. This prediction is complicated by the fact that each arriving job has processing needs on various machines in the shop, while each machine is experiencing different and varying levels of congestion, which in turn change as the job flows through the shop. Moreover, the prediction must include consideration of the particular dispatching scheme in operation in the shop.

However, the order-promising, or due-date-assignment, decision is important as Wein in 1991 [81], Cheng and Gupta in 1989 [14], and others point out. Cheng and Gupta note:

> A review of the literature reveals that this aspect of the scheduling decision [determination of optimal due-date values] is of particular importance to both researchers and practicing managers. Research in this area obviously has not been carried out to its completeness since it is apparent that so many areas still remain untouched. The practicing managers are increasingly faced with difficult situations in which jobs have to be delivered on time, otherwise costs will be incurred. Minimizing inventories is the name of the game in today's extremely competitive business world. It has led to the concept of 'zero inventory' (ZI) which insists on carrying no inventory at all, whether it be raw material or supplies related inventory or finished goods inventory.

Wein adds that "it is very important for due dates to be based on the knowledge of the status of the shop floor and the urgency and importance of the various jobs."

In general, the due-date assignment literature dwells mainly on simple, regression-based approaches for setting the due date, as the 1989 survey article by Cheng and Gupta [14] and an excellent comparative study by Ragatz and Mabert [62] indicate. Initially, researchers examined due-date rules that considered only job characteristics in setting the date; these included TWK, where due dates are based on total work; SLK, where jobs are given flow allowances that reflect equal waiting times or equal slacks; and NOP, where due dates are set according to the number of operations to be performed on the job. Conway [16], Eilon and Chowdhury [20], and Baker and Bertrand [6] conducted such studies. More recently, another class of due-date assignment methods was proposed that includes not only job-characteristic information, but shop-status information as well. As Cheng and Gupta note, many researchers have reported improved performances from these methods, which include: JIQ, JIS, WIQ, and various combinations thereof. All three of those heuristics include the job's total processing time as job-characteristic information, while for the shop-status information JIQ considers current queue lengths (Eilon and Chowdhury [20]), whereas JIS includes the number of jobs in the system (Weeks [79]), and WIQ uses total processing time of all jobs in the workcenter queues on this job's routing when setting the due date.

However, to date, hardly any research has been published using nonlinear versions of these models; i.e., most research has examined only due-date predictions that vary linearly with, for example, jobs in queue, jobs in system, etc. An exception to this is a (short) regional proceedings paper published by Smith and Gee [75] that examines TWK and NOP and their squares, but provides no comparative analysis with linear techniques. Cheng and Gupta conclude that "very little or no work has been done on the dynamic multi-machine problem with sophisticated due date assignment methods."

As mentioned, Ragatz and Mabert [62] published an excellent comprehensive comparison of different due-date assignment rules. As this is the most thorough comparison of due-date rules in the literature, and as there is a need for further comparative research with a standard set of cases, several analyses of this dissertation are based on their framework. They considered the performance of eight different assignment rules in a specific shop which they simulated. Only six of these rules are included in this research, omitting their clearly worst performers, Weeks' rule (see [79]), and another one, JIS, that was almost no different under our shop conditions from others we consider. Furthermore, since there is no comparative study of due-date rules in the literature that includes nonlinearities, a regression model allowing for nonlinear independent variables will, in addition to the inherently nonlinear neural network, also be evaluated in most of the comparisons in this dissertation.

Cost Performance Criteria

For most realistic cases it can be assumed that a shop incurs penalties for early as well as tardy job completion. Unless finished goods can be shipped prior to their respective due dates, early completion will increase monetary and space investments in finished goods inventory. Conversely, tardy completions will oftentimes be even more costly as they produce customer ill will or - more easily quantified, but just as expensive - contract penalties.

The due-date literature has few studies where costs of earliness and tardiness are explicitly considered. Among others, Ragatz and Mabert [62] compared due-date assignment rules on the basis of both mean absolute deviation (MAD) and standard deviation of lateness (SDL) criteria. Such criteria are symmetric in the sense that being early is just as costly as being equally tardy. With MAD the cost function is implicitly linear, whereas with SDL costs are tacitly quadratic. Similar assumptions are implicitly made by the various due-date assignment techniques. For example, practitioners using ordinary least squares (OLS) regression prediction are, whether they recognize it or not, minimizing quadratic cost functions for missing the date; such a technique produces due dates that could result in half or more of the jobs finishing late.

As keeping the customer waiting produces ill will, such assumptions about the cost function can be very costly, as will be demonstrated in this research.

Papers in the scheduling area that do consider costs include one by Weeks and Fryer [80], one by Bookbinder and Noor [11], and a third by Ragatz and Mabert [61]. The Weeks and Fryer paper uses linear and nonlinear regression to estimate the relationship between five response measures and the value of K, the multiple of total processing time employed in assigning due dates. The five response measures are all costs: mean job flowtime cost, mean job lateness cost, mean job earliness cost, mean job due date cost, and mean labor transfer cost. But, as stated, this method only considers one job-specific characteristic, namely total processing time. Shop status data and other job characteristics are not included. Bookbinder and Noor in 1985 looked indirectly at costs by setting due dates subject to service-level constraints. But this work assumed a single-machine shop, and hence is outside our range of interest. Ragatz and Mabert published work in 1988 that looked at costs directly by examining holding and late-delivery costs that were both linear and had varying ratios of the former to the latter in the range of 1:1 to 1:50. However, this work was a comparison of order-release mechanisms, the second of the three production control stages, rather than of due-date prediction policies.

In this research the advantages of directly incorporating more realistic cost functions for early and tardy behavior into the predictive model are shown. Included are cost functions that differ in magnitude and shape on the two sides of the due date. Then, the least total cost due-date prediction techniques are determined for each of the different cost functions examined. By using realistic cost functions rather than other, commonly symmetric criteria in the evaluation of the techniques, the additional costs can be determined that practitioners will incur in practice when using ad hoc rules or simple regression techniques.

Neural Network Literature

Historical Background

The seminal work in neural network research is often taken to be a 1943 paper by Warren McCulloch and Walter Pitts [48]. This paper demonstrated some fundamentals of neural computing and sparked considerable research interest in the field. Further groundwork was laid in 1949 by Donald Hebb [29], who proposed a learning law for the synapses of neurons. Hebb's learning law was originally motivated by experimental results from psychology, but became the starting point for neural network training algorithms.

The first neurocomputer (the Snark) was constructed by Marvin Minsky in 1951 [50]. Although it never actually carried out any interesting information processing function, it did adjust its weights automatically (with motors and potentiometers). Other early advances in neural networks were generated by Frank Rosenblatt ([66] and [67]), Bernard Widrow ([82] and [83]), and others throughout the 1950's and 1960's. The networks they built, called perceptrons, generally consisted of only a single layer of artificial neurons, and were applied to such diverse problems as weather prediction, electrocardiogram analysis, and artificial vision.

However, neural network optimism and activity virtually died down with the publication of the book Perceptrons in 1969 [51]. In this book Minsky and Papert proved theoretically that single-layer networks are incapable of solving many simple problems including the "Exclusive-OR" function. They concluded that the perceptron and neural networks were basically not interesting for further study. Because of their stature in the field, research interest in neural computing was drastically reduced for more than a decade.

It was not until Robert Hecht-Nielsen applied a mathematical proof by Kolmogorov [43] to the neural network arena that interest again began to wax. In particular, Hecht-Nielsen demonstrated that any continuous function can be mapped by a three-layer neural network. This effectively removed the stigma surrounding the technique of being uninteresting and unuseful.

Another potent "force" that helped revitalize research in neural networks in the early 1980's was the work on learning laws and different configurations of neural networks by John Hopfield ([34], [35]), an established physicist. Finally, with the publication of Parallel Distributed Processing by David Rumelhart and James McClelland in 1986 [68], attention in the field of neural networks soared. Over the last few years, research interest and funding has increased drastically, with conferences and journals dedicated solely to neural networks, and the commercialization of this artificial intelligence tool for practical applications has begun.

Models and Applications

Many different neural network models have been developed. Probably the most successful of the existing paradigms, and the one used in this research, is the backpropagation neural network. Backpropagation provides a systematic means for training multilayer networks, thereby overcoming the limitations presented by Minsky and Papert. Rumelhart, Hinton and Williams [69] are among those credited with the development of the backpropagation network.

Backpropagation networks have demonstrated impressive results in many practical applications: Sejnowski and Rosenberg [73] trained a network to convert text to speech; Cottrell, Munro and Zipser [17] designed one for image-compression; Burr [12] reported success in character recognition; Lapedes and Farber [45] successfully applied the algorithm to the prediction of "chaotic" time series. Other applications have been, for example, in the areas of control ([65], [77]), signal processing ([64], [72]), and integrated circuits ([2], [23]).

However, to date, most applications have been to non-managerial scenarios. Among the few studies in the production control area are one by Rakes et al. [63], and another by Wray [85]. These studies use a backpropagation neural network to determine the optimal number of kanbans in a multiline, multistage JIT production system for static and dynamic shop factors, respectively.

In the research reported in this dissertation, a backpropagation neural network is applied to another production problem, namely due-date prediction. The connection weights are modified in the learning process according to a generalized delta rule as outlined by Rumelhart et al. [69]. In chapter 4 a variety of cost functions ~~is~~ are used as the error functions for the neural network. The feedback of error through the network to properly adjust connection weights is modified according to the particular due-date cost function under consideration. The derivation discussed in chapter 4 that shows how to change weights based on the error function, is also predicated on the learning paradigm proposed by Rumelhart. Finally, the neural network training procedure followed throughout this dissertation reflects one suggested by Hinton and Touretzky [33]. This procedure takes special care to avoid overfitting the network in the training phase, a phenomenon called grandmothering.

While backpropagation neural networks appear to be the most popular, many other network paradigms exist, of which only a few are now briefly mentioned. Adaptive Resonance Theory (ART) networks are based upon concepts introduced by Grossberg in 1976 [27]. ART networks are two-layer, nearest-neighbor classifier which are well suited for the classification of input patterns. The similar Bidirectional Associative Memory Model was developed by Kosko in 1988 [44]. It is best applied in situations that require nearest-neighbor pattern matching of only a few patterns. The Brain-State-in-a-Box network was introduced by Anderson et al. in 1977 [3] and is essentially a one-layer, autoassociative, nearest-neighbor classifier with strengths in fault and noise tolerance. The Boltzmann Machine was developed by Hinton et al. in 1984 [32]. It modifies a Hopfield network by using a stochastic update rule, called simulated annealing, to escape from local minima. Its strengths include the ability to perform nonlinear mappings and to analyze combinatorial optimization problems. Finally, Counterpropagation is a network paradigm which was invented by Robert Hecht-Nielsen [30]. It combines a Kohonen layer with another layer employing Grossberg learning, whereby the network functions as a statistically optimal self-programming lookup table.

Overall, the aforementioned paradigms are only a sample of the wide variety of existing neural network models. The purpose of listing them is to engender an appreciation for the versatility and applicability of neural networks. At the same time it is to motivate research examining the applicability of other models to the production area.

Chapter 3

USING NEURAL NETWORKS TO DETERMINE DUE-DATE ASSIGNMENTS

Background

Overview

As first suggested by Baker in 1984 [5] and reiterated by Ragatz and Mabert in 1988 [61], the production control system can be viewed as consisting of three sequential stages: the order promising/master-scheduling stage, the order-release portion, and the shop-floor stage. In stage 1, order-promising/master-scheduling, jobs arrive from the customer and are assigned a due date. If due dates are specified by the production department, they are referred to as "internally-set due dates;" conversely, if due dates are set by an order-entry marketing department in agreement with the customer, they are called "externally-imposed due dates" (see, for example, [62]). This distinction emphasizes the control, or lack thereof, of the production department in its effort to meet its performance objectives. In the second stage, order release, work is sent -- possibly after some intentional delay -- to the shop floor. In the final stage the work itself is performed; during this stage the sequence of work at various constrained workcenters must be determined. This decision is referred to as "dispatching."

Because of the complexity of the production control system, the literature has typically not examined all three stages in toto. Likewise, in this chapter we concentrate on the first stage of the production control system, namely order promising. In particular we focus on the prediction of internally-set due dates. In so doing, we assume that there is no intentional delay in the second stage, order release. (I.e., we use "immediate order release.") Moreover, we only consider the case where dispatching in the third stage is according to a shortest processing time (SPT) rule, and no early deliveries are permitted (although, of course, early completion of jobs is allowed, but not preferred). Finally, we examine only Eilon's 1978 [19] dynamic, multi-machine case, where by dynamic is meant that the number of jobs available for processing varies over time, as opposed to the static case where all jobs are available for processing at one starting time.

The order-promising, or due-date-assignment, decision is important as Wein in 1991 [81], Cheng and Gupta in 1989 [14], and others point out. Cheng and Gupta note:

> A review of the literature reveals that this aspect of the scheduling decision [determination of optimal due-date values] is of particular importance to both researchers and practicing managers. Research in this area obviously has not been carried out to its completeness since it is apparent that so many areas still remain untouched. The practicing managers are increasingly faced with difficult situations in which jobs have to be delivered on time, otherwise costs will be incurred. Minimizing inventories is the name of the game in today's extremely competitive business world. It has led to the concept of 'zero inventory' (ZI) which insists on carrying no inventory at all, whether it be raw material or supplies related inventory or finished goods inventory.

Wein adds that "it is very important for due dates to be based on the knowledge of the status of the shop floor and the urgency and importance of the various jobs."

Assigning due dates is also a difficult decision. As a job arrives at the shop, a due date is predicted indicating when the job is expected to complete. This prediction is complicated by the fact that each arriving job has processing needs on various machines in the shop, and by the further fact that each machine is experiencing different and varying levels of congestion, which change as the job flows through the shop. Moreover, the prediction must include consideration of the particular dispatching scheme in operation in the shop. As mentioned, SPT dispatching is assumed in this chapter. This dispatching rule is particularly difficult to predict with, as jobs already in the shop will lose their places in queues to later-arriving jobs with smaller processing times. Therefore, although we focus on due-date assignment, order dispatching is an integral part of the decision; we are merely restricting ourselves to one important case of shop-floor control, SPT dispatching.

In this chapter we consider a new procedure for internally setting due dates, namely, neural network prediction. Neural networks are an artificial intelligence (AI) approach that has been applied to such general problem areas as prediction, control, data compression, and surface fitting (see, for example, Klimasauskas [42]). However, most applications to date have been to non-managerial scenarios such as robot control, visual systems, airport bomb detection, etc. Here, we attempt to see whether neural networks can outperform conventional, regression-based due-date assignment rules taken from the literature.

Our purpose is to begin the analysis and discussion of neural network applicability in this area. It is difficult to predict how neural networks will perform versus established rules across a variety of shops and with different sample sizes, given the complexity of the due-date decision process. Since, as will be pointed out, the literature is so sparse in the area of nonlinear models altogether, we add to one of the shops used in our comparison a regression case where we allow nonlinearities in the variables as well. Therefore, the research in this chapter considers two additions to the literature: neural networks and regression with nonlinear models. The reader should be cautioned, however, that no claim is made here that we have examined the myriad number of nonlinear regression models or, for that matter, the many different neural network implementations possible. Our contribution is that the neural network implementation used here surpasses the rules used in the literature in a statistically significant manner. This is true whether the criterion is either of two widely-used measures in the due-date literature, mean absolute deviation or standard deviation of lateness. Further, it is demonstrated in this chapter that additional study of nonlinear regression methods is needed as well, as those methods outperform conventional rules also for the shop examined.

The remainder of this chapter is organized as follows. The rest of this background section first discusses the particular rules for due-date assignment we compete against (summarized in [62]), then presents the nonlinear regression case we also consider, and finally concludes with a brief excursus on neural network basics for the uninitiated reader. This is followed next by a section on the procedure used in our comparison of methodologies and then by a section on the respective results. Upon finding that the neural network is a statistically significant better predictor than the conventional (regression-based) rules, two sections describing extensions to the standard experiment are presented: first an investigation of the effect of sample size, and then an inquiry into the effect of shop structure on our initial results. Finally, the last section provides conclusions.

Conventional Due-Date Assignment Rules

The due-date assignment literature for the dynamic, multi-machine case dwells mainly on simple, regression-based approaches for setting the due date, as the 1989 survey article by Cheng and Gupta [14] and an excellent comparative study by Ragatz and

Mabert [62] indicate. Initially, researchers examined due-date rules that considered only job characteristics in setting the date; these included TWK, where due dates are based on total work; SLK, where jobs are given flow allowances that reflect equal waiting times or equal slacks; and NOP, where due dates are set according to the number of operations to be performed on the job. Conway [16], Eilon and Chowdhury [20], and Baker and Bertrand [6] conducted such studies. More recently, another class of due-date assignment methods was proposed that includes not only job-characteristic information, but shop-status information as well. As Cheng and Gupta note, many researchers have reported improved performances from these methods, which include: JIQ, JIS, WIQ, and various combinations thereof. All three of those heuristics include the job's total processing time as job-characteristic information, while for the shop-status information JIQ considers current queue lengths (Eilon and Chowdhury [20]), whereas JIS includes the number of jobs in the system (Weeks [79]), and WIQ uses total processing time of all jobs in the workcenter queues on this job's routing when setting the due date.

However, to date, hardly any research has been published using nonlinear versions of these models; i.e., most research has examined only due-date predictions that vary linearly with, for example, jobs in queue, jobs in system, etc. An exception to this is a (short) regional proceedings paper published by Smith and Gee [75] that examines TWK and NOP and their squares, but provides no comparative analysis with linear techniques. Cheng and Gupta conclude that "very little or no work has been done on the dynamic multi-machine problem with sophisticated due date assignment methods."

As mentioned, Ragatz and Mabert [62] published an excellent comprehensive comparison of different due-date assignment rules. As this is the most thorough comparison of due-date rules in the literature, and as we see a need for further comparative research with a standard set of cases, we base our study on their framework. They considered the performance of eight different assignment rules in a specific shop which they simulated. We compare our neural network against six of these rules, not considering their clearly worst performances (Weeks' rule, see [79]) and one (JIS) that was almost no different under our shop conditions from others we consider. Moreover, one of the shops we study here is the same shop studied by Ragatz and Mabert.

The remaining six rules can be segmented into two categories: those rules containing only information on the job whose due date is to be determined, and those rules that also include data on shop status at the time of arrival of the job whose scheduled completion time is to be ascertained. Using the notation in [62], the six rules used in our comparison are as follows:

Rules using job characteristics only:

1. Total Work (TWK):

   $$\text{Flow}_i = kP_i$$

   The estimated flow time of job i (Flow_i) is a function of the total processing time P_i and a parameter k which must be estimated.

2. Number of Operations (NOP):

   $$\text{Flow}_i = kN_i$$

   Here the estimated flow time of the job is a function of the number of operations (N_i) on job i's routing. Once again, k is a (different) parameter that must be estimated.

3. Total Work and Number of Operations (TWK + NOP):

   $$\text{Flow}_i = k_1P_i + k_2N_i$$

   Both number of operations and processing times are used in this rule.

Rules using job characteristics and shop data:

4. Jobs in Queue (JIQ):

$$\text{Flow}_i = k_1 P_i + k_2(JIQ_i)$$

At the instant job i is to be released to the shop, all work centers on job i's path through the shop are polled and the total number of jobs in queues there are summed (JIQ_i). This piece of shop data is combined with the job characteristic P_i.

5. Work in Queue (WIQ):

$$\text{Flow}_i = k_1 P_i + k_2(WIQ_i)$$

This rule differs from rule 4 in that the total processing time (rather than just the <u>number</u> of jobs) on job i's path through the shop is used in the formula.

6. Response Mapping Rule (RMR):

$$\text{Flow}_i = f(X_{i1}, X_{i2}, ..., X_{in})$$

This approach was originally suggested by Ragatz and Mabert; in it, important independent variables are identified, and then various functional rule equations are estimated. The basic concept with this rule is that the flow time through the shop is a complex function of many shop and job characteristics; identifying the proper set for a given shop and then including these variables explicitly in flow time estimates should improve accuracy of prediction.

Ragatz and Mabert concluded that due-date rules based on job-characteristic information alone performed more poorly than those rules also containing shop-status information, but that more complex rules did not necessarily do any better than simpler rules, when both contained job characteristic and shop status data. In contrast to this final conclusion, we hope to demonstrate that more detailed information does in fact provide better due-date predictions than just aggregate information when processed by a neural network.

Nonlinear RMR Regression

Since there is no comparative test of due-date rules in the literature that includes nonlinearities, we specify such a model for our comparison. As it is not our primary purpose to study nonlinear regression, but rather to examine neural network applicability, we take a single nonlinear case. For the case chosen, we will show superior behavior to the conventional models, thereby indicating further study of such cases is needed, the point we hoped to make.

The model tested here is a nonlinear version of the RMR rule mentioned above. In particular,

$$\text{Flow}_i = g(X_{i1}, X_{i2}, \ldots, X_{in})$$

where the function g is second order in any of the variables (or their crossproducts) determined in the (linear) RMR case. We call this new prediction method "nonlinear RMR regression."

Excursus: Neural Network Basics

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

There are many different types of neural networks, including Hopfield, Brain-State-in-a-Box, Bidirectional Associative Memory, Boltzmann, Adaptive Resonance Theory, Hamming, and Spatiotemporal Networks ([31], [42], [53], [74], [78]). Probably the most common and readily available of the neural networks is the backpropagation network. This network has been used with success in many applications including forecasting [42], the area of interest here. Because the standard backpropagation network has achieved forecasting success in other domains, and because it is probably the most widely known neural network, we use this particular network in our study. This choice is consonant with one of our ultimate objectives, to make advantageous artificial intelligence approaches usable for practitioners.

The operation of a single neuron is modeled as a processing element (PE) in a backpropagation neural network. The PE may have many inputs (coming from sensory inputs, other neurons, or alternatively from the output of this neurode looping back on itself). Just as with the brain, each of these inputs is separately attenuated by a weight W_{ij}; it is this weight that represents the synaptic gap. All weighted inputs are then combined by the PE, typically by a simple summation function, i.e., $I_i = \sum_j W_{ij} X_j$. The processing element then generates an output that is a simple function of the input; a sigmoid function such as a hyperbolic tangent is often used as the output transfer function. In this research, the output function (y) chosen is another sigmoid function, $y = 1/(1 + \exp(-I_i))$, where I_i is the simple summation function defined above. This logistic sigmoid function is also one of the most commonly used transfer, or so-called, "squashing" functions (see, for example, [31], [74], [78]). It should be noted that the way a processing element learns is by modification of its weights (the W_{ij}).

As mentioned, a backpropagation network is used in this research. Backpropagation networks consist of an input layer, an output layer, and one or more "hidden" (middle) layers of processing elements. These networks are characterized by supervised learning, whereby training data are used to "teach" the network. In particular, each presentation of training data will include a set of inputs and the corresponding desired set of outputs. If the actual output obtained from the network by presentation of the training inputs differs from the desired output, then the weights interconnecting the neurodes are adjusted in a fashion to bring the actual output closer in line to the desired output. A particular training data set often must be presented tens or hundreds of times to the network before the underlying relationship between inputs and desired outputs is learned.

Deciding when to stop training is not so easy, since neural networks often overtrain (so-called "grandmothering") in that they learn more than just the functional relationship underlying the data; they learn the peculiarities and vagaries of the particular data set used to train. Therefore, in a solidly-designed study, a second data set is utilized to decide when to stop training. This is done by evaluating the performance of the trained (by the first data set) network progressively at different numbers of training cycles on the fresh (second) data set. The training then is stopped when the trained network produces best performance on the second data set.

Once a neural network is trained, it is "ready" to be presented with additional data, including data never seen by the network before. This is called the recall phase. In this phase, training is over, and all weights are frozen. An input vector of interest may be applied to the network, from which the network's output is then determined algebraically.

Additional basic information on neural networks and backpropagation is available, for example, in Rumelhart and McClelland [68], Klimasauskas [42], Hecht-Nielsen [31], and Müller and Reinhardt [53].

Procedure

To compare the neural network due-date prediction approach and the nonlinear model approach with the six conventional rules' predictions, a five-step procedure was implemented. In the first step, a SLAM II [59] simulation model of a shop was written and then validated. In the second step 3,500 jobs were generated, "pushed" through the shop, and data were collected on them. These data were used in the next step to estimate parameters in the six rules in the conventional case as well as in the nonlinear regression case and to provide training and stopping data to determine weights for the neural network case. In the fourth step another 10,000 new jobs were generated and processed by the shop. These data were then used in the final step to statistically compare the performance of the two methodologies according to mean-absolute-deviation and standard-deviation-of-lateness criteria. This procedure will now be explained in more detail. All steps of the procedure were implemented on an IBM 3090 (mainframe) computer, using extensive software written by the author.

Step 1: The Shop Simulation

The shop chosen for study is indicated in Figure 3-1. Jobs entering the shop were routed through the five workcenters in a sequential order. At each workcenter the jobs were processed at one of two available machines, the determination of which was made for each job when it entered the shop.

Machine processing times were drawn from a negative exponential distribution with a mean of 1.8. Job interarrival times were also selected from a negative exponential distribution, but with a mean of 1.0. In addition, an order-release delay of zero (i.e., no delay) was assumed when releasing jobs to the shop. Finally, the shortest processing time (SPT) dispatching rule was used in all cases discussed in this chapter.

Step 2: Data Generation for Regression Modeling and Network Training

Both individual-job-characteristic data and shop-status data were collected for 3,500 jobs generated in this step of the methodology. In particular, for each job, eight general, job- and shop-characteristic data were obtained as well as three workcenter-specific data at each of the five workcenters, resulting in 8+(3*5)=23 characteristics per job (see Table 3-1). In addition, the actual flow time through the shop was observed, from which the due date could be determined. Except for the actual flow time, all data generated were collected when the job entered the shop. As a result, all shop-status data are reflective of the condition of the shop at the instant of job entry, as is desired.

Two points should be noted with respect to the generation of these 3,500 jobs. First, the jobs to be included in the sample of size 3,500 were generated only after a

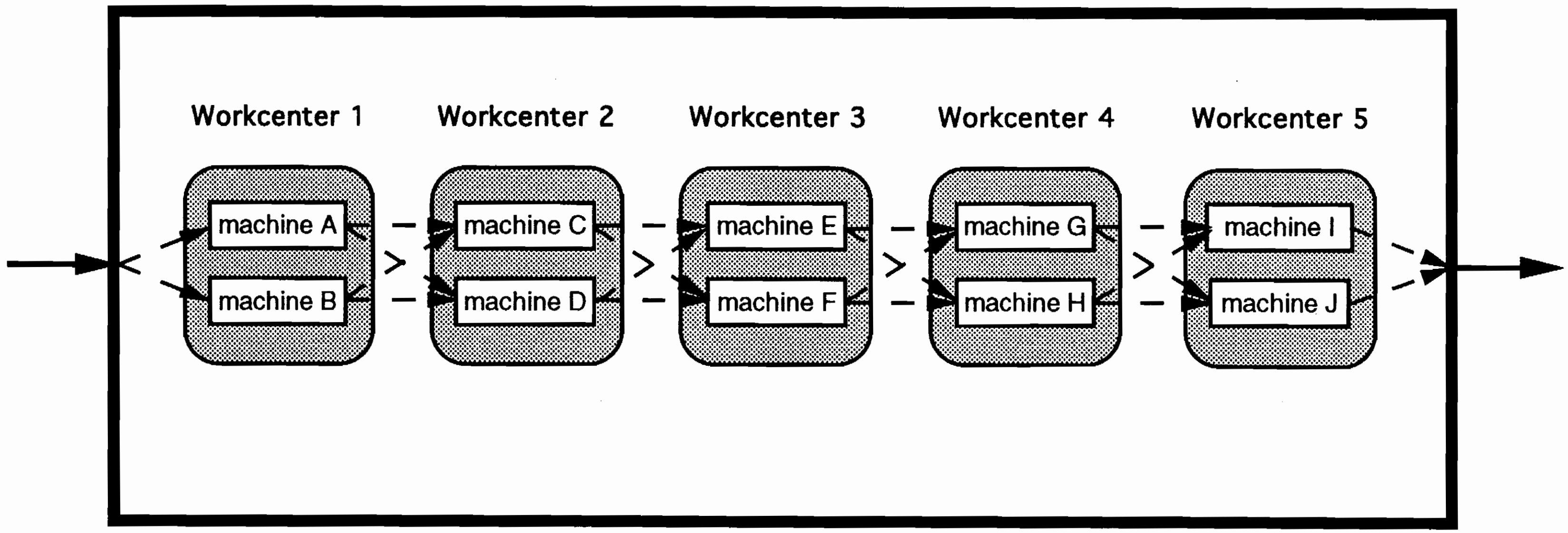


Figure 3-1: The shop used to test the methodologies

Table 3-1: The 23 inputs for each job j

Input	Information
	General Shop Characteristics
1	number of operations required by job j
2	sum of processing times for job j
3	number of operations that must be done by the machines on job j's path to complete all jobs presently in the shop
4	sum of jobs presently in queues on job j's routing
5	maximum processing time job j requires on any machine
6	number of operations required to empty the shop of its current workload
7	sum of the expected waiting times at machines on job j's routing
8	total processing time of jobs in queues on job j's routing
	Workcenter-Specific Characteristics
9,...,13	processing time for operation 1,...,5
14,...,18	number of operations which must be done on the machine which does operation 1,...,5 on job j in order to complete all the jobs which are presently in the shop
19,...,23	number of jobs presently in queue on the machine which does operation 1,...,5 on job j

significant warm-up period of 300 data points had transpired, allowing the shop to be fully operational before sample data were collected. Second, as will be seen below in the neural network training procedure, the 3,500 jobs were collected in two groups, one of size 2,800 jobs and another of length 700. A period of length 1,000 jobs wherein no data were collected was allowed to pass between these two groups to reduce the possibility of serial correlation between them. Since we wanted both the conventional cases and the neural networks to see precisely the same data, the regression cases were presented the same 3,500 data points as the AI approach, even though there was no need for the gap of 1,000 in those cases.

Step 3a: Conventional Rules' Regression Modeling

All 3,500 data points generated in step 2 were next used to estimate the parameters of an ordinary least squares regression model for each of the six linear and the one nonlinear rule described above. The SAS software package [70] was utilized.

Step 3b: Neural Network Training

A backpropagation network with 23 input processing elements (also known as nodes), 9 hidden layer PEs, and 1 output PE was utilized for this phase of the study and is indicated in Figure 3-2. Each of the 23 input nodes was supplied with one of the 23 data characteristics generated for each job, with the one output node representing the due date. The bias node shown in that figure provides a constant (equal to one) input and acts much like an intercept term in a regression model; bias is universally included in backpropagation networks.

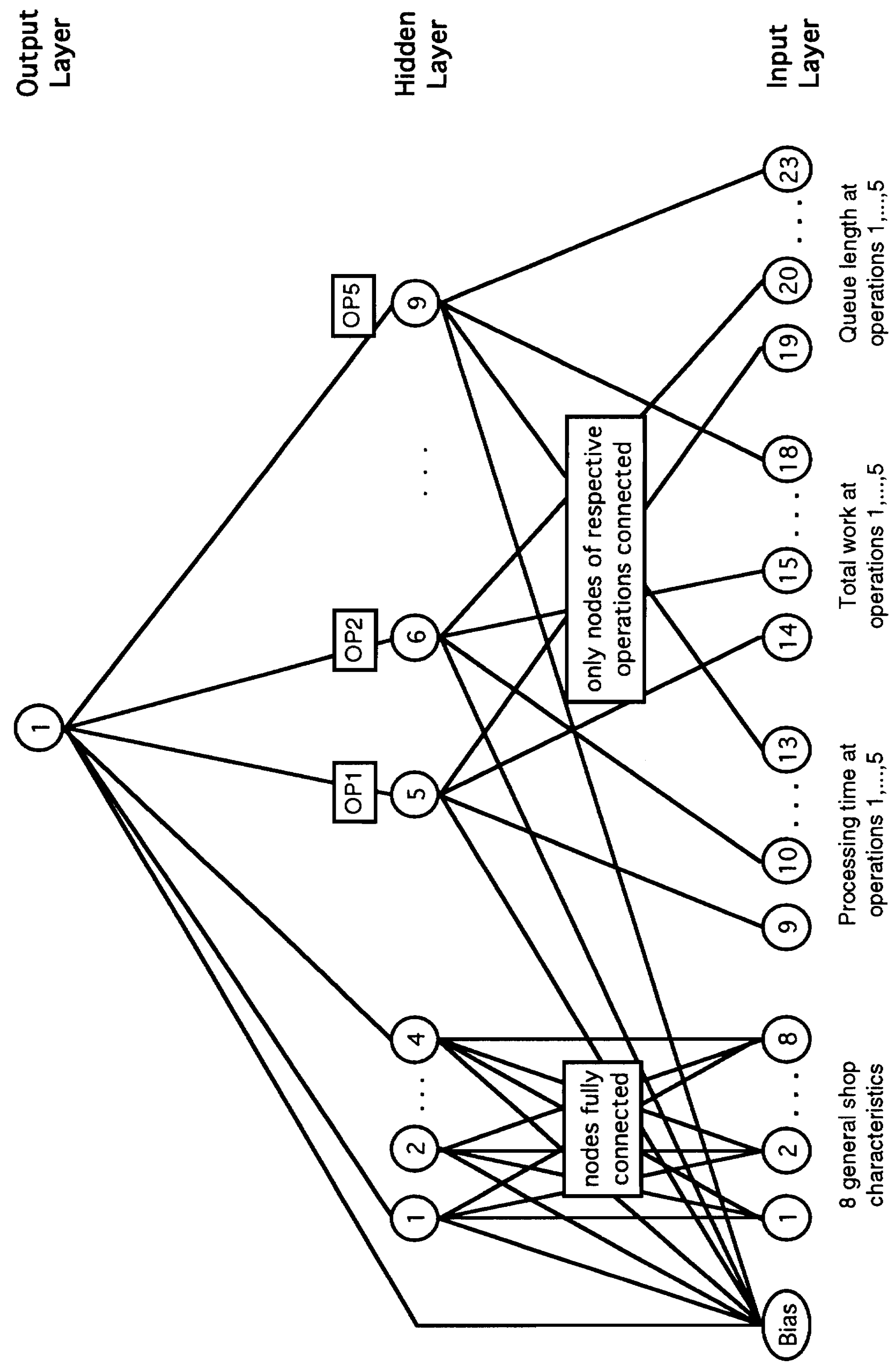

Output Layer
Hidden Layer
Input Layer
1
OP1
OP2
OP5
1
2
4
5
6
9
only nodes of respective operations connected
nodes fully connected
Bias
1
2
8
9
10
13
14
15
18
19
20
23
8 general shop characteristics
Processing time at operations 1,...,5
Total work at operations 1,...,5
Queue length at operations 1,...,5

Figure 3-2: The backpropagation network used for due-date prediction

Generally, in a backpropagation network, all PEs in the input and output layers are fully connected to the hidden layer, and the number of PEs in the hidden layer is chosen by a rule of thumb such as "use half the sum of the number of input and output nodes," which in this case would have been 0.5*(23+1)=12. Only nine hidden layer nodes were used here, however, and the PEs were not fully connected because shop configuration information was utilized implicitly, thereby reducing complexity of the neural network, as will now be explained. As shown in Figure 3-2, eight of the input nodes representing general characteristics were fully connected to the first four hidden layer nodes, as is common practice. However, each of the other five hidden layer nodes was only connected to three of the input nodes. This is because there is no reason to expect activities at one operation to significantly affect those at another, given the randomness with which the jobs were routed through the shop. Therefore, hidden layer node 5, for example, was used to learn about the first operation (only), as its only three inputs were nodes 9 (the processing time for the first operation for this job), 14 (the anticipated total time to clear the system at this operation) and 19 (the current number of jobs in queue at this operation).

In the conventional rules as in the nonlinear model the 3,500 data points generated in step 2 were used to estimate the models' parameters. In the neural network case, these 3,500 points were used to both train the network and then decide when to stop training. We arbitrarily allocated 80% of the 3,500 data points (i.e., 2,800) to training, while reserving the other 20% (700 points) to decide when to stop the training process. This explains why we had generated two independent data sets in step 2; we wanted the training and the stopping data to be as little correlated as possible. The specific procedure of determining the optimal number of training cycles was as follows: We first trained the neural network for 25 training cycles on the 2,800 data points (i.e., 25*2,800 presentations to the network), then we froze the weights and applied the other 700 data points to the network to determine performance according to the criterion (say, MAD). The presentations of data points during training and stopping were all made in random order, to reduce serial correlation. This gave us the first data point plotted in Figure 3-3. We then retrained the network with 50 presentations of all 2,800 data points, froze the new weights and reapplied the 700 data points to again determine performance. This process was repeated until an entire curve such as that of Figure 3-3 was produced. Then the minimum point on the curve was chosen as the point to stop training, which in the case given is 175 training cycles. Once the optimal training point was determined, we retrained the entire network with all 3,500 points, again randomly presented, (not wanting to waste any data,) stopping at 175 training cycles.

Recall that in a solidly-designed study, a second ("stopping") data set should be used to decide when to stop training; if this is not done, so-called "grandmothering" can occur as can be seen in conjunction with Figure 3-3. The figure indicates that performance on the fresh, never-before-seen data set improves initially with more training, then deteriorates with additional training beyond 175 cycles. However, if a stopping data set is not used, but rather the training data set itself is employed to decide when to stop training, the modeler can be misled by supposedly increasingly better performance of the neural network with the number of cycles. In fact, doing so (deciding when to stop training with our network on the same data that it had been trained on) would have suggested better performance on up to roughly 500-1,000 cycles before "leveling off." As Figure 3-3 proves, such a result would have been erroneous. The neural network

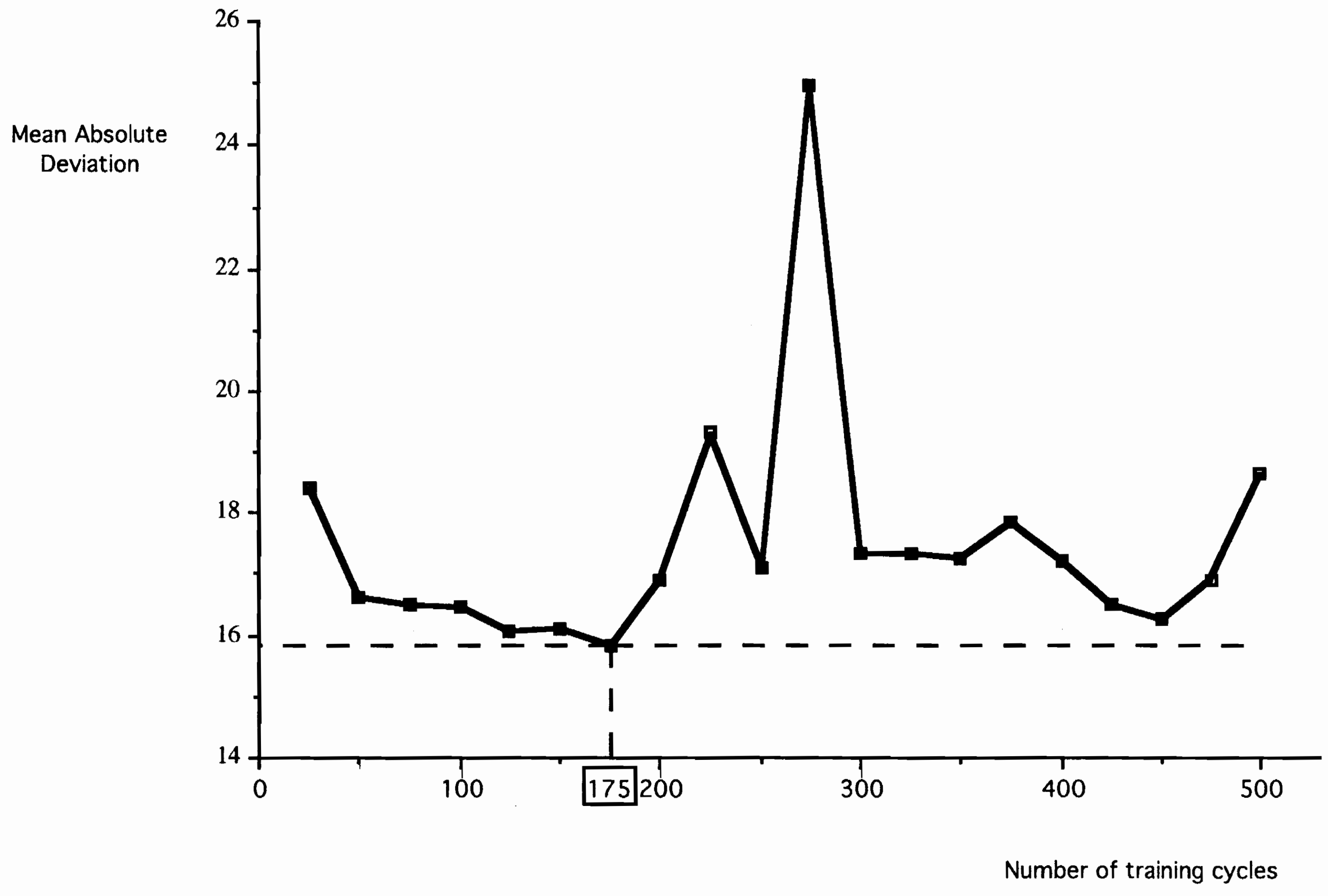


Figure 3-3: Determining the number of cycles to train, utilizing 20% of the sample size (3,500) as stopping data

would really just have been memorizing better the data it had seen, not improving its ability to generalize.

Step 4: Data Generation for Testing both Methodologies

This step is similar to the procedure of step 2 except that now data on 10,000 jobs were collected from the simulation model. The conventional rules as well as the neural network were then tested on those data in the following, final (5th) step.

As these data are to be used in testing for statistical significance, it is necessary to guarantee statistical independence among the data before the test is performed. As a first step toward insuring independence of these data, only one in every 50 outputs from the shop simulation, on average, was included in the sample of 10,000 jobs. Stated differently, there was only a 2% chance that any given job would be included in the sample in order to reduce the serial correlation.

Step 5: Statistical Significance Test

Once all parameters for the regression methods had been determined in step 3, and all the neural network weights had been set in the same phase, all parameters and weights were frozen and tested using the additional 10,000 data points generated in step 4 from the same shop. The same 10,000 points were presented to the conventional rules as to the neural network weights; hence, a paired statistical test (the paired-t test) was suggested.

Since the 10,000 points might still be serially correlated, thus violating the independence assumption of the paired-t test, the "batch-means method" (see, for example, Law and Kelton [46]) was used. In particular, 10 batches of size 1,000 each were formed. Since the ten batch means were serially "far apart" from each other, they could safely be assumed to be statistically independent. Further recall that since only one population exists in the paired-t test (that of the differences), there is no need to test for homoskedasticity of variances.

As mentioned, two frequently used criteria from the literature were also used here: the mean absolute deviation (MAD) from the desired due date, and the standard deviation of lateness (SDL) of the predicted due date. We formulated the one-sided hypothesis that the MAD for the six rules in the conventional approach does not exceed that of the neural network, a hypothesis we hoped to reject with statistical significance. In particular, we tested:

$$H_0: \mu_D \leq 0$$

$$H_1: \mu_D > 0,$$

where D is the difference between the conventional MAD and the neural network MAD. Then the same hypothesis was tested using SDL rather than MAD as a criterion. For both tests we set the confidence level at α=0.01, using Bonferroni's inequality (see, for example [56]) to control the experimentwise error rate of the multiple comparisons.

Results

In summary, the procedure described above was followed to generate 3,500 data points from the shop shown in Figure 3-1; those data points were used to estimate parameters in the six rules of the conventional as well as in the nonlinear regression case and the PE weights in the neural network case. Then 10,000 test points were generated to test whether the neural network approach is significantly better than any of the six conventional rules. As mentioned above, the 10,000 test points were grouped into ten batches; the batch means are indicated in Table 3-2 for all the conventional cases (the first six columns), the nonlinear RMR case (labelled NLN) and the neural network case (NN) for each of the two criteria, MAD and SDL.

The data of Table 3-2 indicate that the null hypothesis above is clearly rejected for both the mean-absolute-deviation as well as the standard-deviation-of-lateness criterion at the confidence level of 0.01, whereby the experimentwise error rate is controlled by using Bonferroni's inequality (i.e., each pairwise comparison was tested at an error rate of 0.01/6). In fact, the pairwise comparisons of the neural network with each of the six conventional rules all gave p-values of less than 0.0005 for either the MAD or the SDL criterion. The interpretation of these results is that the neural network outperforms all six, conventional rules examined here in a statistically significant manner.

These findings give rise to hopes of using this AI methodology as a preferred means of estimating due dates. However, extreme caution must be exercised as only one shop has been examined (so far), and that at (only) a sample size of 3,500 data points.

Furthermore, Table 3-2 also demonstrates the superior performance of NLN (nonlinear RMR regression) over the six conventional linear rules in the literature, although the NLN results are not as favorable as the neural network's. This excellent NLN performance strongly suggests that regression nonlinear in its variables also is worthy of further study for the due-date assignment problem.

This chapter of the dissertation proceeds as follows. As neural networks are sometimes accused of being "data hungry," i.e., they only work well on large-sized samples, in the

Table 3-2: Batch data and t-values of the paired-t tests for 3,500 sample points in the standard shop

Mean Absolute Deviation

Batch	TWK	NOP	TWK+NOP	JIQ	WIQ	RMR	NLN	NN
1	17.0	20.1	17.0	17.1	17.0	16.1	13.5	13.2
2	18.3	22.8	17.7	19.0	19.0	17.5	15.2	15.0
3	20.9	25.6	20.4	21.8	21.9	20.7	18.9	18.5
4	18.6	22.0	18.3	19.2	19.1	18.0	15.7	15.2
5	18.5	22.1	18.3	18.9	18.9	17.6	15.9	14.8
6	21.3	25.1	20.8	21.9	22.0	20.7	19.1	18.1
7	19.6	23.3	19.2	20.2	20.3	19.1	17.4	15.9
8	18.2	20.9	17.7	18.8	18.7	17.9	15.8	14.8
9	18.0	21.4	17.4	18.4	18.3	16.9	15.1	14.0
10	18.0	21.7	17.6	18.6	18.6	17.3	15.9	14.9

The t-values of the pairwise comparison of the neural network with each conventional rule:

	TWK	NOP	TWK+NOP	JIQ	WIQ	RMR	NLN
NN	t = 24.6	t = 50.4	t = 18.1	t = 40.6	t = 43.1	t = 27.9	t = 6.2

Standard Deviation of Lateness

Batch	TWK	NOP	TWK+NOP	JIQ	WIQ	RMR	NLN	NN
1	34.1	37.6	34.2	33.9	33.9	32.7	31.7	32.6
2	45.1	49.6	44.6	45.0	45.0	43.0	41.3	40.6
3	58.2	62.5	57.8	58.2	58.1	56.4	55.7	55.5
4	39.9	45.0	39.4	40.0	40.0	38.2	36.5	36.2
5	43.1	47.3	42.8	42.9	42.9	41.0	39.8	37.2
6	52.1	56.8	51.6	52.1	52.1	50.6	50.2	49.4
7	44.4	50.0	43.6	44.5	44.5	42.0	40.9	37.7
8	36.1	40.8	35.7	36.2	36.2	35.2	34.5	32.4
9	34.0	38.9	33.7	34.0	34.0	32.3	31.5	30.8
10	34.8	40.0	34.5	34.9	34.9	33.5	33.1	32.2

The t-values of the pairwise comparison of the neural network with each conventional rule:

	TWK	NOP	TWK+NOP	JIQ	WIQ	RMR	NLN
NN	t = 7.4	t = 13.7	t = 7.2	t = 7.3	t = 7.3	t = 4.8	t = 2.7

following section we first examine the same hypotheses on the same shop, but with differing sample sizes. Then, in the subsequent section, we conduct two more experiments, first with a "highly structured" shop, and finally with a very "unstructured shop" (i.e., unstructured in that routings through the shop are completely random).

Effect of Sample Size

The exact same procedure as described above was again followed on the shop of Figure 3-1. However, this time sample sizes of magnitude 500, 1,000, 2,000, 5,000, and 6,500 were generated in step 2 and used for regression modeling and neural network training in step 3. The curves exhibited in Figures 3-4 and 3-5 were generated.

In these figures the criterion (MAD or SDL) is plotted versus sample size. At the top of each figure are the plots of the conventional rules suggesting their poor performance relative to the neural network across the entire range of sample sizes. Conducting the hypothesis test at each sample size verifies this expectation, with statistically significant differences found at each size for an experimentwise significance level of 0.01. Furthermore, an improved performance of the nonlinear model compared to the linear conventional rules is shown, while in both sets of curves at all sample points neural networks performed better than the nonlinear method considered.

One additional point should be mentioned with respect to these two figures. Note that several observations are indicated at sample size values of 500, 1,000, and 2,000; each of these observations represents an additional, independent input sample of new data

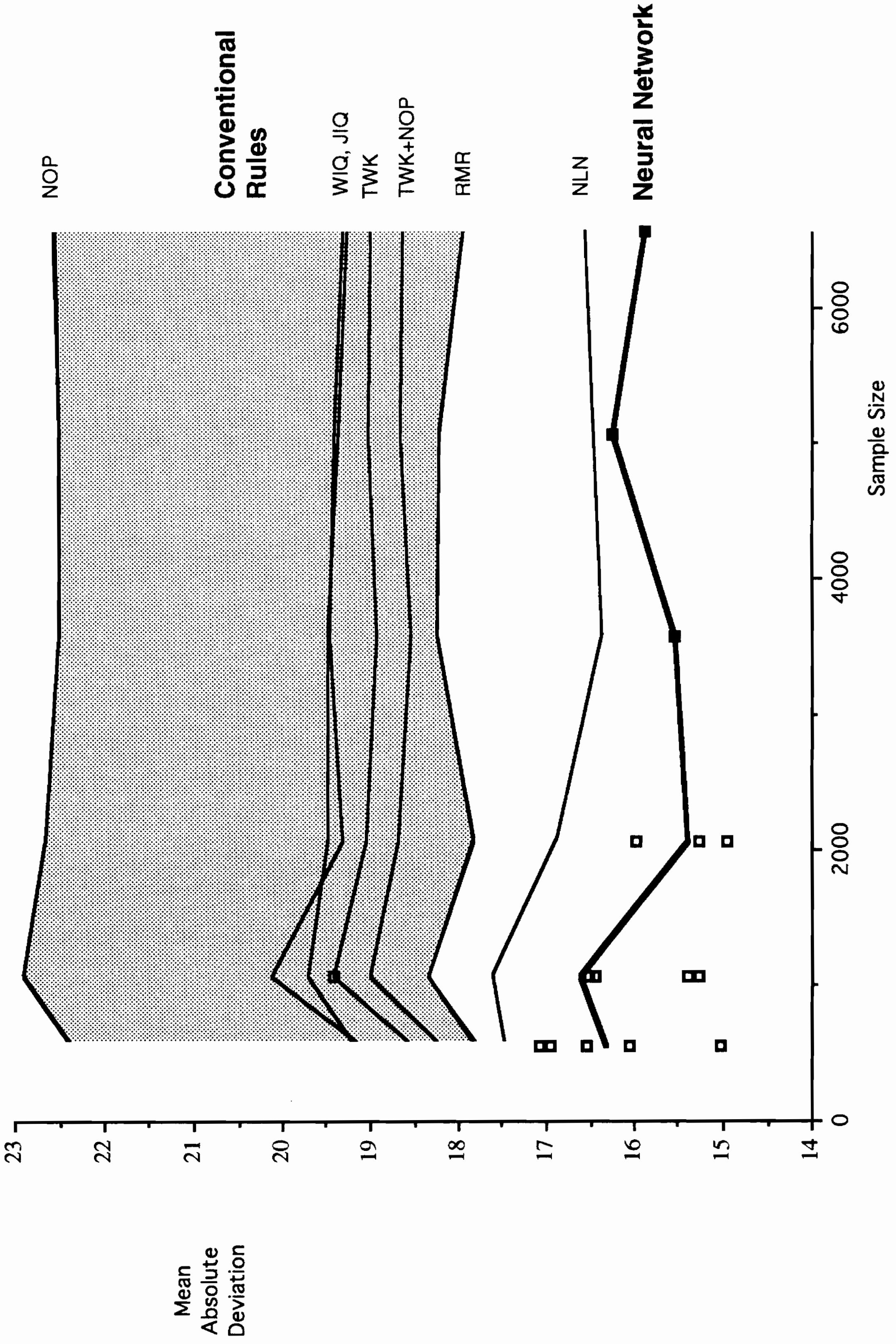


Figure 3-4: Effect of sample size on MAD for the standard shop

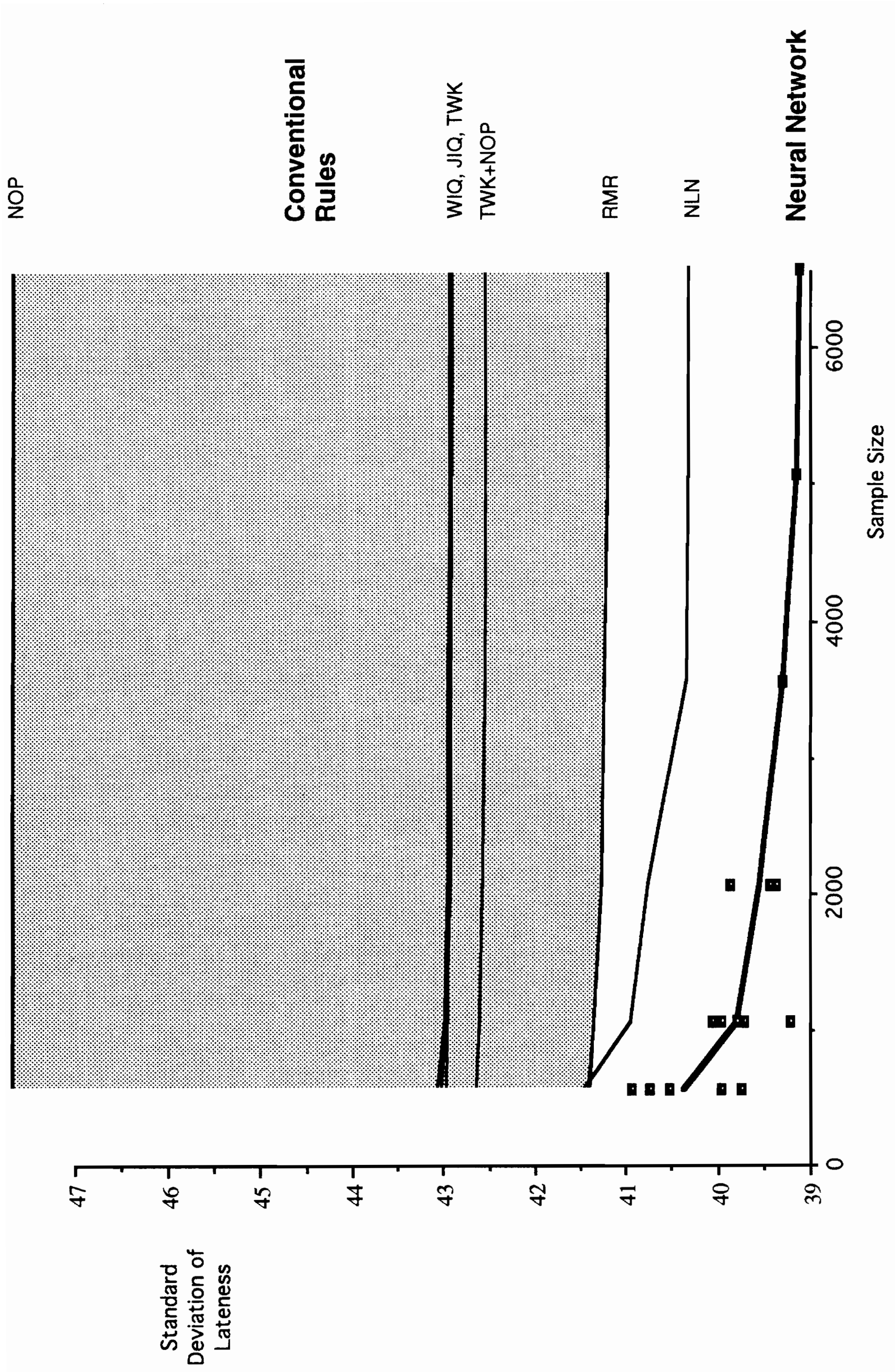


Figure 3-5: Effect of sample size on SDL for the standard shop

generated from the shop with which the neural network was then re-fitted and tested again. The point to be made from this testing is that there is variation across samples presented to the neural network, and that this variation is larger than the variability found with similar extra samples (not shown in the figures) used to re-fit the conventional rules. However, in every instance, each neural network observation was significantly better than its conventional counterparts.

Effect of Shop Structure

Having determined that neural networks provided the methodology of choice regardless of sample size for the shop of Figure 3-1, our attention turned to an investigation of the effect of shop structure. Naturally, one could not hope to generalize results across all shops because of the wide range of operating conditions in "real-world" environments. Rather, we more modestly investigate two further shops, one with considerably more structure than that of Figure 3-1, and one with much less structure. We investigate the performance of the linear rules studied in the literature versus that of the neural network. We claim no generalizability of our results beyond the three shops studied.

The "highly structured" shop we studied next differs from that of Figure 3-1 in that it has only one machine at each workcenter. All jobs now have a fixed, common routing through the five machines/workcenters of the shop. Processing and interarrival times were adjusted to keep the utilization of the shop approximately the same as in the baseline case. The same neural network as shown in Figure 3-2 was used, and the same procedure as described above was followed.

Results are indicated in Figures 3-6 and 3-7. Once again the neural network outperforms the conventional rules. Hypothesis testing again indicates statistical significance at an experimentwise error rate of 0.01. Note, however, that although comparison of these figures with Figures 3-4 and 3-5 is somewhat dangerous because processing and interarrival times had to be adjusted as explained, both the MAD and SDL are lower in the more-structured shop, as expected.

The final case chosen for study was an "unstructured" shop, which is essentially the shop studied in Ragatz and Mabert [62]. This shop differed from that of Figure 3-1 in that 15 workcenters were now present and random routings to any of the 15 were allowed. In addition, the number of operations performed on each job ranged from 1 to 15, drawn from a geometric distribution. (Once again parameters had to be adjusted to keep utilization comparable across shops.) The random routings combined with the shortest processing time (SPT) dispatching rule, we believed, would make it difficult for any procedure (or person) to predict due dates because knowledge of jobs not-yet-arrived would be critical. To see this, consider a case where a job arrives which requires a particularly long operation time for its last operation which is at, say, workcenter 1. Regardless of the shop status, it is impossible to know when this job will complete at workcenter 1, much less leave the shop, if there are jobs that arrive at the shop subsequently with work to be done at workcenter 1. This is because they will "bump" the job in question because of their relatively short processing times and the SPT rule. The problem is exacerbated by the fact that all routings are random, so many jobs can jump in front of this one. We therefore questioned the ability of either any conventional or the AI-based method to do well in this shop with SPT dispatching.

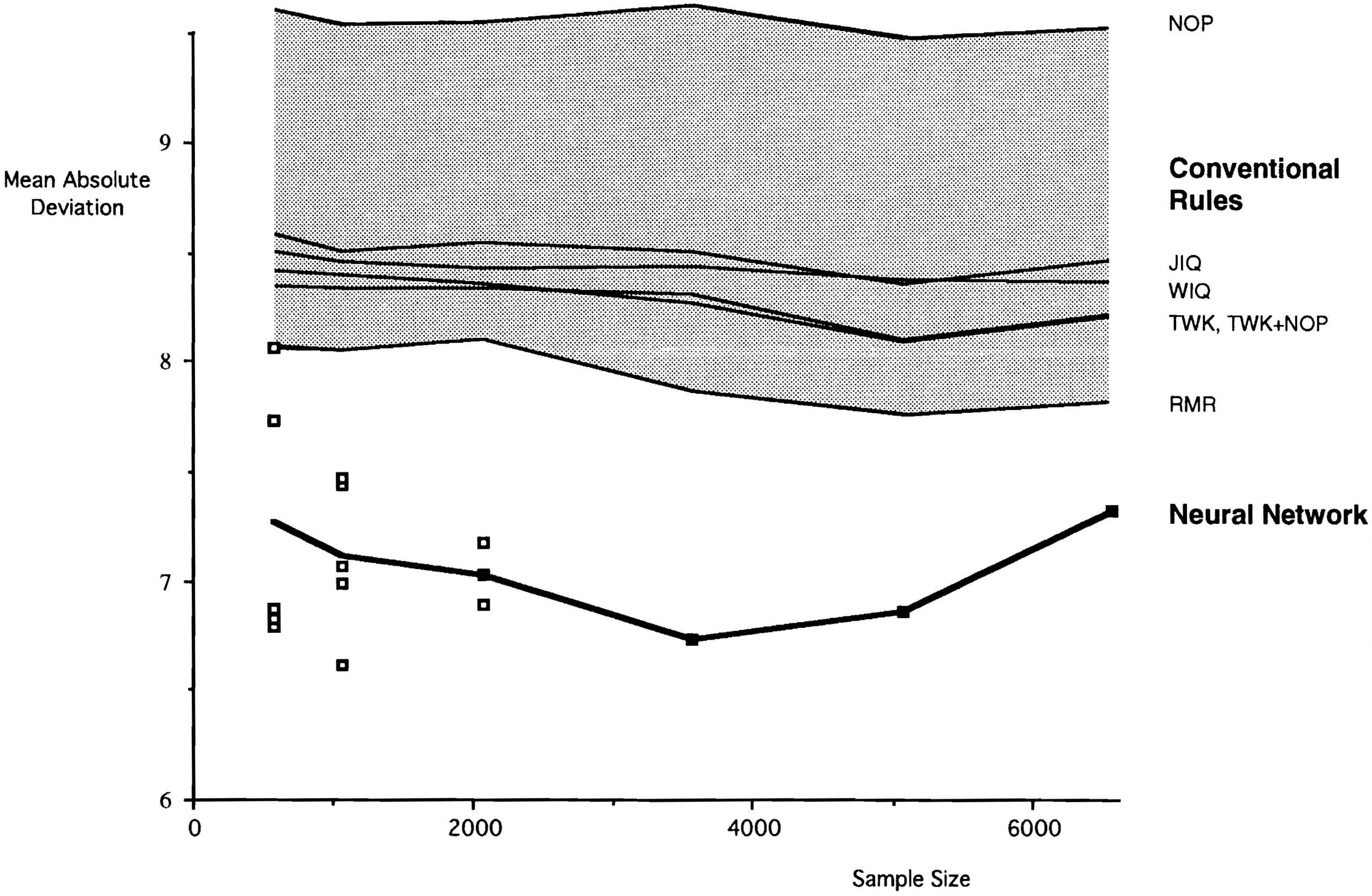


Figure 3-6: Effect of sample size on MAD for the highly structured shop

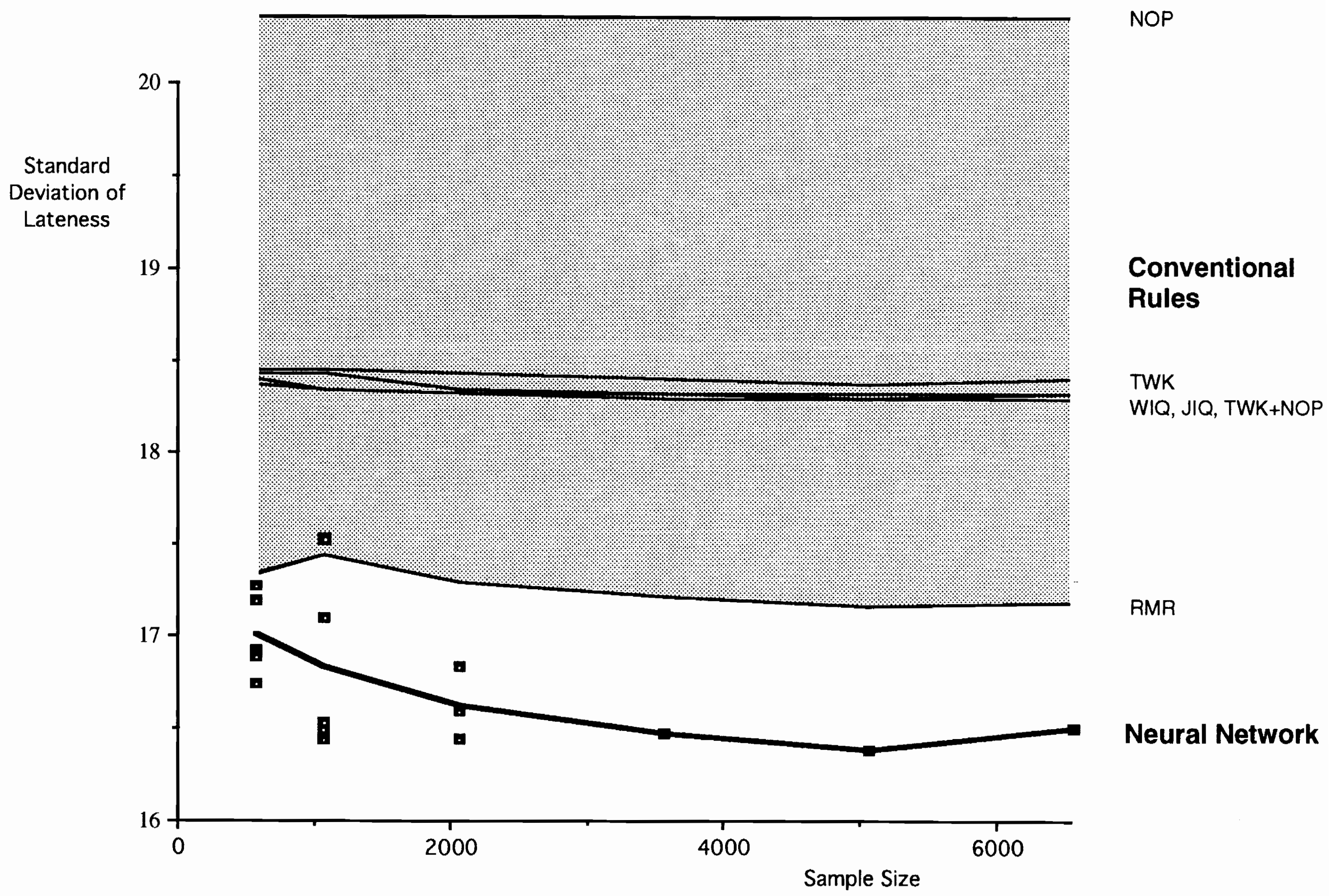


Figure 3-7: Effect of sample size on SDL for the highly structured shop

The procedure is the same as before, except that now there will be 15 operations (rather than 5) represented in the input and hidden layers. There are therefore 53 (i.e., 8+(3*15)) input nodes and 19 (i.e., 4+15) hidden layer PEs. The number of output nodes remains one. Results from this case, shown in Figures 3-8 and 3-9, are different than before, but are generally unsurprising. It is seen from these plots that the neural network does not "lose" universally to the conventional methods, but neither does it outperform those approaches on both criteria across all sample sizes. It is our belief that there is too much unpredictability in the shop for either method to do well or to dominate the other. If the shop were to be made more structured (e.g., by either making the routings less random or by changing the dispatching rule from SPT to, say, first-come, first-served), we conjecture that the neural network would again do better. This topic we leave for further study.

Conclusions

It is the general conclusion of this chapter of the dissertation that neural networks are worthy of further experimentation as the methodology of choice for due-date prediction. It was found that in two of three shops studied, the neural network outperformed conventional statistical methods, and that it did no worse overall on the third.

Furthermore, it is concluded that regression models nonlinear in their independent variables also are worthy of detailed examination, as they too outperformed the (linear) conventional rules cited in the literature. This chapter therefore has demonstrated superior performance of two due-date prediction approaches to those cited in the literature.

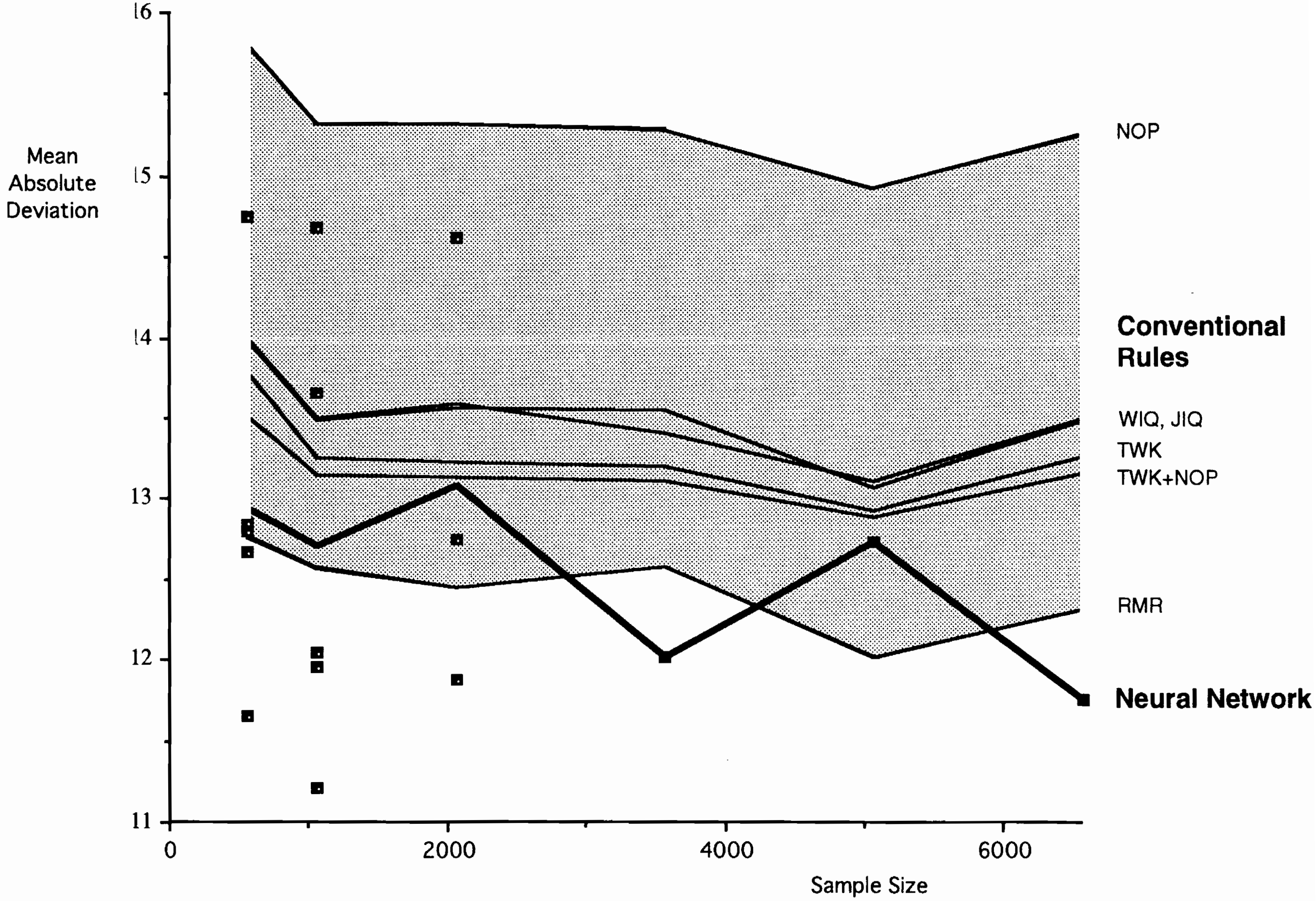


Figure 3-8: Effect of sample size on MAD for the highly unstructured shop of Ragatz & Mabert

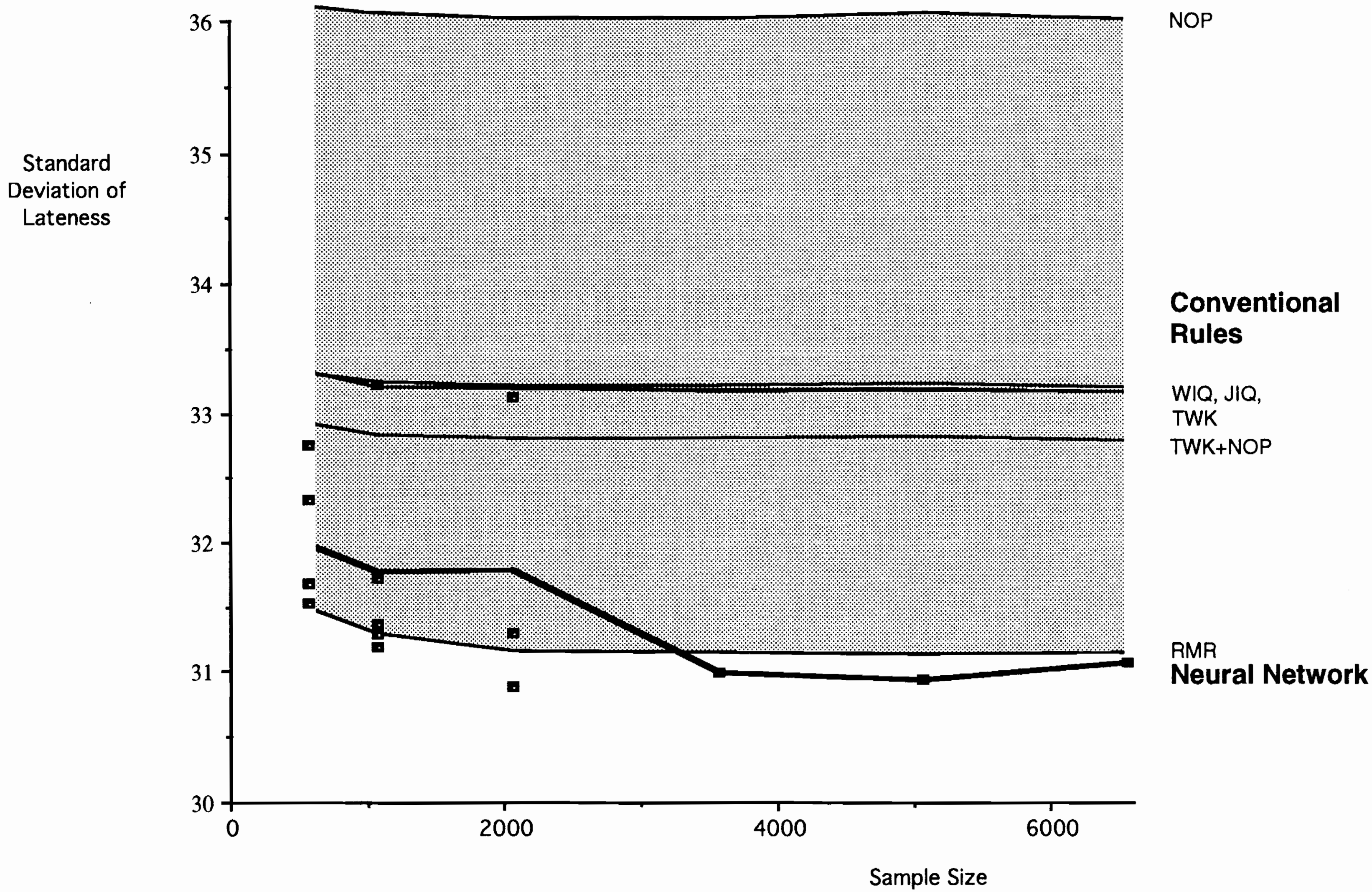


Figure 3-9: Effect of sample size on SDL for the highly unstructured shop of Ragatz & Mabert

Ragatz and Mabert concluded in their study [62] that more complex rules using more job and shop information do not necessarily perform any better than simpler rules with less information. We have found that neural networks using all job and shop information available outperform regression rules with less information about the orders. But what we have not demonstrated yet is whether neural networks with more job information perform better than neural networks with less such data. We now briefly describe the results of a small study we pursued to examine this issue.

We first trained a new neural network only on the few inputs used by the RMR method, the best performer of the conventional rules. We then compared its performance on 10,000 recall data points (generated as described above) against the performance of our full, "all data" neural network model of Figure 3-2. Indeed, the neural network using more data did better than the one using less data. We repeated the same test for different sample sizes in the standard shop, and all results unanimously support the conclusion that more data are better than less data (by approximately 5% on average) if neural networks are used.

As a check, we repeated the same comparison of less versus more explanatory variables for a regression-based model. The results of this effort supported the results found earlier by Ragatz and Mabert that, contrary to the neural network case, more job and shop information did not help the regression performance. We therefore conclude that with neural networks, more detailed information is preferred to less detailed information, in the cases considered. This amends Ragatz and Mabert's finding that using more data does not necessarily do any better than simpler rules with less data to read "in the case of conventional, regression-based rules."

This modification to Ragatz and Mabert's finding is also consonant with Robert Hecht-Nielsen's statement [31] that

> In essence, in terms of its everyday practice, there has only been modest progress in regression analysis since the days of Gauss. Neurocomputing is now providing a breath of fresh air to this 200 year old subject.
>
> ...[E]nough experimental evidence has now been gathered to state with some confidence that [neural] mapping networks are, in general, different than statistical regression approaches. The function approximations that arise from properly applied mapping networks ... are usually better than those provided by regression techniques .... This difference is particularly important in high-dimensional spaces (input dimensions greater than 3 to 10), where many of the more "automated" regression techniques often fail to produce an appropriate approximation. [Emphasis added].

One additional important topic that we have not discussed thus far is the effect of multicollinearity, which certainly is present in the neural network model of Figure 3-2 due to the correlation between some of its inputs. Although we are not prepared to offer a definitive statement on this matter, it is our tentative conclusion that multicollinearity has a much smaller deleterious effect on neural network mapping than it has on multiple regression. We cite two reasons for our belief. First, theoretical work reported by I. S. Markham [47] concludes that neural networks perform much better than regression in the presence of correlated input data. However, her test cases were markedly simpler than our full-data scenarios. Thus, secondly, we again conducted a small experiment to examine the effects of the correlation among inputs 1-8 in Figure 3-2 with inputs 9-23. We removed this correlation by dropping inputs 9-23 altogether in a new neural network configuration for the standard shop. We noted over several sample sizes that inclusion of the correlated data provided no detrimental effect. If our tentative conclusion proves correct that multicollinearity is less troublesome for neural networks than for regression, then this would certainly be a decided advantage of the former technique. We suggest that additional work is warranted.

Another important issue, and the final one discussed here, involves the practical implications of our study, in particular, whether neural networks can be used by practitioners. As we have discovered superior performance of the neural network approach in due-date prediction, we believe it is incumbent on us to at least touch upon this topic. In general, we do not foresee any issue that will prohibit implementation by practitioners. Backpropagation neural network codes are available from a variety of vendors (for example [42]). From a sample size standpoint, we have generally suggested that the more data collected, the better the neural network performance will be up to a point, even though relatively small samples appear to suffice for prediction improvements over conventional methods. It is true, however, that more data per sample is better than less data, so it would behoove a practitioner to collect as much data across the shop as possible. In this regard, practitioners with lower-dimension input spaces would perhaps do well to delay implementation and wait for further research data to be collected for that case, following the implied caution emphasized in Hecht-Nielsen's comments cited above. The specific neural network model used would depend on the particular case, but probably would not need to vary much from the architecture proposed in Figure 3-2. The training and stopping procedures would be carried out as we have outlined. The neural network performance could then be tested (off-line) on new jobs, as they come in. Once the performance is deemed acceptable, the neural network could be brought on-line. Of course, there are many issues that need further study, such as what is the best way to update (in a Bayesian sense) existing weights once a network has been made operational. But we do not believe any issue to be prohibitive to "real-world" application.

We think that the next stage of research in the due-date assignment area should also explicitly include cost-based comparisons of the techniques available to practitioners. That is, we believe the cost of early completion of jobs should be explicitly specified as a function of earliness, and the cost function for tardiness should also be stipulated, as it is likely to differ markedly in form from that for earliness. Then the performance of various rules and techniques, including regression with nonlinear models and neural networks, should be ascertained. We suspect that the cost differences with respect to conventional rules will be large, given the statistically significant differences in performance we have found in this chapter. This topic is studied in detail in the next chapter of this dissertation.

Chapter 4

COST-BASED DUE-DATE PREDICTION USING CLASSICAL AND NEURAL NETWORK APPROACHES

Introduction

Overview

As first suggested by Baker in 1984 [5] and reiterated by Ragatz and Mabert in 1988 [61], the production control system can be viewed as consisting of three sequential stages: the order promising/master-scheduling stage, the order-release portion, and the shop-floor stage. In stage 1, order-promising/master-scheduling, jobs arrive from the customer and are assigned a due date. If due dates are specified by the production department, they are referred to as "internally-set due dates;" conversely, if due dates are set by an order-entry marketing department in agreement with the customer, they are called "externally-imposed due dates" (see, for example, [62]). This distinction emphasizes the control, or lack thereof, of the production department in its effort to meet its performance objectives. In the second stage, order release, work is sent -- possibly after some intentional delay -- to the shop floor. In the final stage the work itself is performed; during this stage the sequence of work at various constrained workcenters must be determined. This decision is referred to as "dispatching."

Because of the complexity of the production control system, the literature has typically not examined all three stages in toto. Likewise, in this chapter we, again, concentrate on the first stage of the production control system, namely order promising. In particular we focus on the prediction of internally-set due dates. In so doing, we assume that there is no intentional delay in the second stage, order release (i.e., we use "immediate order release"). Moreover, we only consider the case where dispatching in the third stage is according to a shortest processing time (SPT) rule, and no early deliveries are permitted (although, of course, early completion of jobs is allowed, but not preferred). Finally, we examine only Eilon's 1978 [19] dynamic, multi-machine case, where by dynamic is meant that the number of jobs available for processing varies over time, as opposed to the static case where all jobs are available for processing at one starting time.

The order-promising, or due-date-assignment, decision is important as Wein in 1991 [81], Cheng and Gupta in 1989 [14], and others point out. Cheng and Gupta note:

> A review of the literature reveals that this aspect of the scheduling decision [determination of optimal due-date values] is of particular importance to both researchers and practicing managers. Research in this area obviously has not been carried out to its completeness since it is apparent that so many areas still remain untouched. The practicing managers are increasingly faced with difficult situations in which jobs have to be delivered on time, otherwise costs will be incurred. Minimizing inventories is the name of the game in today's extremely competitive business world. It has led to the concept of 'zero inventory' (ZI) which insists on carrying no inventory at all, whether it be raw material or supplies related inventory or finished goods inventory.

Wein adds that "it is very important for due dates to be based on the knowledge of the status of the shop floor and the urgency and importance of the various jobs."

The due-date assignment literature for the dynamic, multi-machine case dwells mainly on simple, regression-based approaches for setting the due date, as the 1989 survey article by Cheng and Gupta and an excellent comparative study by Ragatz and Mabert [62] indicate. Initially, researchers examined due-date rules that considered only job characteristics in setting the date; these included TWK, where due dates are based on total work; SLK, where jobs are given flow allowances that reflect equal waiting times or equal slacks; and NOP, where due dates are set according to the number of operations to be performed on the job. Conway [16], Eilon and Chowdhury [20], and Baker and Bertrand [6] conducted such studies. More recently, another class of due-date assignment methods was proposed that includes not only job-characteristic information, but shop-status information as well. As Cheng and Gupta note, many researchers have reported improved performance from these methods, which include: JIQ, JIS, WIQ, and various combinations thereof. All three of those heuristics include the job's total processing time as job-characteristic information, while for the shop-status information JIQ considers current queue lengths (Eilon and Chowdhury [20]), whereas JIS includes the number of jobs in the system (Weeks [79]), and WIQ uses total processing time of all jobs in the workcenter queues on this job's routing when setting the due date.

However, to date, hardly any research has been published using nonlinear versions of these models; i.e., most research has examined only due-date predictions that vary linearly with, for example, jobs in queue, jobs in system, etc. An exception to this is a (short) regional proceedings paper published by Smith and Gee [75] that examines TWK and NOP and their squares. Cheng and Gupta conclude that "very little or no work has been done on the dynamic multi-machine problem with sophisticated due date assignment methods."

The due-date literature also has few studies where costs of earliness and tardiness are explicitly considered. For example, Ragatz and Mabert compared due-date assignment rules on the basis of both mean absolute deviation (MAD) and standard deviation of lateness (SDL) criteria (the same criteria were used in chapter 3). Such criteria are symmetric in the sense that being early is just as costly as being equally tardy. With MAD the cost function is implicitly linear, whereas with SDL costs are tacitly quadratic. Practitioners using ordinary least squares (OLS) regression prediction are thus, whether they recognize it or not, minimizing quadratic cost functions for missing the date; such a technique produces due dates that could result in half or more of the jobs finishing late. As keeping the customer waiting often produces ill will, such assumption can be very costly, as will be demonstrated later in the chapter.

Papers in the scheduling area that do consider costs include one by Weeks and Fryer [80], one by Bookbinder and Noor [11], and a third by Ragatz and Mabert [61]. The Weeks and Fryer paper uses linear and nonlinear regression to estimate the relationship between five response measures and the value of K, the multiple of total processing time employed in assigning due dates. The five response measures are all costs: mean job flowtime cost, mean job lateness cost, mean job earliness cost, mean job due date cost, and mean labor transfer cost. But, as stated, this method only considers one job-specific characteristic, namely total processing time. Shop status data and other job characteristics are not included. Bookbinder and Noor in 1985 looked indirectly at costs by setting due dates subject to service-level constraints. But this work assumed a single-machine shop, and hence is outside our range of interest. Ragatz and Mabert published work in 1988 that looked at costs directly by examining holding and late-delivery costs that were both linear and had varying ratios of the former to the latter in the range of 1:1 to 1:50. However, this work was a comparison of order-release mechanisms, the second of the three production control stages, rather than of due-date prediction policies.

The objective of this chapter is to show the advantages of directly incorporating more realistic cost functions for early and tardy behavior into the predictive model. Included are cost functions that differ in magnitude and shape on the two sides of the due date. Moreover, a second purpose is to determine the least total cost due-date prediction technique for each of the different cost functions examined. Meeting these two objectives will allow us to note the additional costs incurred when, for example, ad hoc rules or simple regression techniques are used in practice.

In our examination of best technique, we consider linear programming, quadratic programming, OLS regression, and supervised (backpropagation) neural networks. In the OLS regression case, we explore models both linear and nonlinear in job-characteristic and shop-status variables, an extension of the existing literature. We compare results using all these methodologies on a sample shop both against each other and also against six other regression-based rules mentioned in the literature.

Contributions of this chapter include introducing cost-based considerations directly into the evaluation of best technique for the multiple-machine, dynamic due-date assignment case; showing how an artificial intelligence technique, neural networks, can be used successfully in this scenario; and suggesting that omitting nonlinear terms in regression-based due-date settings is costly.

The chapter is organized as follows. In the next section, the development of the cost-based approach for neural networks and the classical techniques is presented. This will be straightforward for the classical techniques, but more involved for neural networks. The third section pursues the question of "best" technique and details the procedure used to compare the different approaches. Results of the experiments are furnished in the following section, and the chapter finishes in section five with a statement of conclusions and a discussion of future work.

Three Approaches to Cost-Based Due-Date Prediction

In this section of chapter 4 three solution approaches to cost-based due-date prediction are formulated: those based on neural networks, those utilizing mathematical programming (both linear and quadratic programming are considered), and those incorporating OLS regression. As mathematical programming and regression are well known, they will be dealt with summarily. As neural networks are less familiar to practitioners and researchers in the scheduling area, they will be dealt with in more detail than is common. In fact, a brief overview of neural network basics is given before the cost-based formulation is presented.

Neural Networks

Neural networks consist of neurodes, also called processing elements (or PEs), whose performance is ostensibly based upon the operation of neurons in the brain. Each neurode includes a "weight," which is similar to the synaptic gap in the brain in the sense that each changes as learning takes place, and in that "knowledge" is stored in synapses in the brain and in the weights of neural networks.

Processing elements are interconnected into neural networks. There are many possible network configurations of PEs. One of the most common, and the network used in this research, is the backpropagation network. Backpropagation networks consist of an input layer, an output layer, and one or more "hidden" (middle) layers of processing elements.

These networks are used for supervised learning, whereby training data are used to "teach" the network. In particular, each presentation of training data will include a set of inputs and the corresponding set of desired outputs. If the actual output obtained from the network by presentation of the training inputs differs from the desired output, then the weights interconnecting the neurodes are adjusted in a fashion to bring the actual output closer in line to the desired output. A particular training data set often must be presented tens or hundreds of times to the network before the underlying relationship between inputs and outputs is learned.

Once a neural network is trained, it is "ready" to be presented with additional data, including data never seen by the network before. This is called the recall phase. In this phase, training is over, and all weights are frozen. An input vector of interest may be applied to the network, from which the network's output is then determined algebraically.

Additional basic information on neural networks and backpropagation is available, for example, in Rumelhart and McClelland [68], Klimasauskas [42], Hecht-Nielsen [31], and Müller and Reinhardt [53].

The specific manner in which weights are adjusted in backpropagation networks is now presented in overview fashion. This will lead directly into the derivation of the cost-based due-date prediction equations for backpropagation neural networks.

Data are presented to a neural network at the nodes in the input layer, as shown at the bottom of Figure 4-1; these data are not modified by the input nodes. The data are then sent to the nodes at the hidden layer along the connections between the input and hidden layers. Three things happen at the hidden layer nodes, as indicated in the figure by the

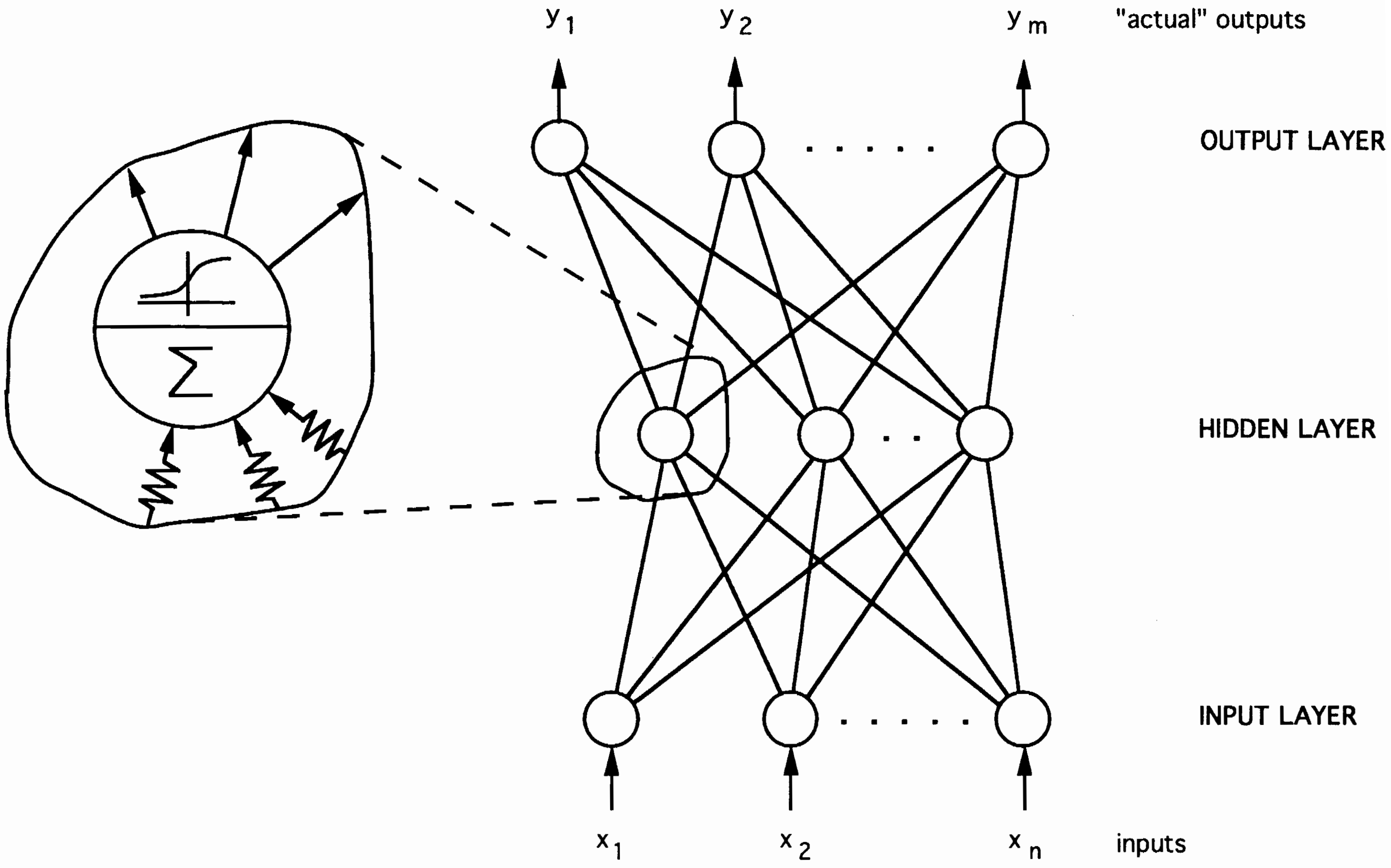


Figure 4-1: Forward pass in a simple backpropagation network

exploded view of one of these nodes. First, each input is attenuated or amplified by the weight at the hidden layer, where there is a (usually different) weight for each connection from the input layer. Next, all the modified inputs appearing at each hidden layer node are summed. Third, the sum at each hidden layer PE is passed through a nonlinear transfer function; we will use the logistic sigmoid function. The output of this function forms the output of each hidden layer node. Then the outputs of the hidden nodes are passed on to the top layer, along the connections between the two layers, where the same three things happen (but now using the output layer weights). The output values at the output layer are the "actual output" of the neural network. The whole process described so far is called the "forward pass" of the network.

Next, the actual output of the neural network is subtracted from the known, desired output to define an error that will be used to modify the weights. For our application a predicted due date is reflected by the actual network output, the completion time of the respective job is given as the desired output, and the resulting error quantifies how early or tardy the job is finished. The error function utilized in backpropagation networks depends on problem characteristics. Here we stipulate that our error function will be the cost function of missing the due date: the tardiness function for positive values of error, and the earliness function for negative error. The error as modified by the cost function is then propagated back through the network (during the so-called "backward pass") as will be explained momentarily, and the weights are adjusted to map inputs into outputs more accurately the next time. As mentioned, training involves tens or hundreds of presentations of a data set to a network, where a forward pass, a calculation of error, and a backward pass must be completed for each presentation of each item in the training data set.

The derivation of neural network weight changes as a function of cost is now developed. For pedagogical reasons and since the neural networks used in this dissertation have only one output node, assume there is only one neurode in the output layer. Further assume there are h neurodes in the hidden layer. The following equations may then be written describing the three activities that occur at the output layer neurode (call it node 1).

Let x_j = the input from the j^{th} hidden layer neurode

w_{1j} = the weight at output node 1 along the connection from hidden layer node j

y_1 = the actual output at the output layer neurode

d_1 = the desired output at the output neurode

e_1 = $d_1 - y_1$ = the error at output node 1

C = the total cost, a function of the error at output node 1.

Then, to model attenuation/amplification and summation at output node 1, let

$$z_1 = \sum_{j=0}^{h} w_{1j}\, x_j. \qquad (1)$$

Here the j = 0 term represents a connection to a bias node (common in backpropagation networks), where $x_0 = 1$. The general, nonlinear transfer function may be written as $\underline{y} = f(\underline{z})$, or in our case using the logistic sigmoid transfer function

$$y_1 = f(z_1) = (1 + \exp(-z_1))^{-1}. \qquad (2)$$

Now the objective in our formulation is to change the weights in the network in such a manner as to minimize the cost function C at the output layer neurode:

$$\Delta w_{1j} = \frac{-\partial C}{\partial w_{1j}}.$$

By using the chain rule,

$$\Delta w_{1j} = \frac{-\partial C}{\partial w_{1j}} = \frac{-\partial C}{\partial y_1} * \frac{\partial y_1}{\partial z_1} * \frac{\partial z_1}{\partial w_{1j}}. \qquad (3)$$

Explicitly introducing the cost function $C = g\ (d_1 - y_1) = C(e_1)$ gives

$$\frac{\partial C}{\partial y_1} = g'. \qquad (4)$$

Note here that since the weights should not be modified when there is no error, the value of C when $e_1 = 0$ is not important to us.

Then from (2), $\frac{\partial y_1}{\partial z_1}$ is just the partial derivative of the transfer function. Moreover, if the logistic sigmoid is used as the transfer function, then it is easy to show that

$$\frac{\partial y_1}{\partial z_1} = y_1(1-y_1). \qquad (5)$$

Finally, from equation (1)

$$\frac{\partial z_1}{\partial w_{1j}} = x_j. \qquad (6)$$

Now substituting equations (4), (5), and (6) into equation (3), we obtain

$$\Delta w_{1j} = -g' * y_1(1-y_1) * x_j, \qquad (7)$$

which specifies how to change the weights as a function of the cost function, the result of interest.

As will be explained in detail later, we will use as our earliness and tardiness cost functions polynomials of degree three or less. (This is because only simpler cost functions can be solved using the classical solution techniques, mathematical programming and regression; this is as opposed to neural networks, which can be utilized with any differentiable cost function.) Then the derivative g' is also a simple polynomial. Note that in the case where a linear cost function is used, a somewhat counterintuitive result is obtained, namely that, because g' is a constant, weight changes are independent of the magnitude of the error. Consequently, for the absolute value cost function evaluated later, weight changes only depend on the sign of the errors and not their magnitude.

Similar kinds of manipulations as have just been demonstrated for the output layer need also be performed to calculate the weight changes at the hidden layer.

Once the weights are determined in this fashion, i.e., once the neural network is completely trained, the weights are frozen. Then for any application of inputs x, the output y_1 may be found algebraically by first applying the hidden layer equivalents of equations (1) and (2) at that layer, before utilizing equations (1) and (2) at the output layer.

Mathematical Programming

The linear programming model is simply

$$\text{Min} \sum_{i=1}^{N} d_i^+ + \sum_{i=1}^{N} d_i^- \qquad (8)$$

subject to

$$\beta_0 + \sum_{j=1}^{m} \beta_j x_{ij} - y_i + d_i^+ - d_i^- = 0, \qquad i=1,\ldots,N \qquad (9)$$

$$d_i^+,\ d_i^- \geq 0,\ \ \beta_j \text{ unrestricted.} \qquad (10)$$

Here the ß's represent the (m+1) coefficients to be estimated when fitting the prediction straight line, the y's the actual completion times, the d_i^+ the tardiness of the i^{th} sample data point, and the d_i^- the earliness of the i^{th} point. Once again the x's represent the data input. N is the total number of data points used to estimate the best straight line.

Note that the constraints (9) each express the truism that:

predicted due date - actual completion date + tardiness - earliness = 0.

Further note that a job cannot simultaneously be tardy and early.

Later we also consider a piecewise linear cost function where it is ten times as costly to be tardy as to be early. In this case objective function (8) is modified to be

$$\text{Min} \ \ (10\sum_{i=1}^{N} d_i^+ + \sum_{i=1}^{N} d_i^-) .$$

Similarly, the quadratic programming model is

$$\text{Min} \ \sum_{i=0}^{N} (d_i^+)^2 + \sum_{i=0}^{N} (d_i^-)^2 \qquad (11)$$

subject to the same constraints ((9) and (10)) as before.

Again, later in the chapter we also consider a linear earliness cost function combined with a quadratic tardiness cost function. In this case the objective function becomes

$$\text{Min} \ \sum_{i=0}^{N} (d_i^+)^2 + \sum_{i=0}^{N} (d_i^-) .$$

Once the appropriate mathematical programming model has been solved for the (m+1) values of the ß's, they are then used in the prediction equation

$$\hat{y} = ß_0^* + \sum_{j=1}^{m} ß_j^* x_{ij},$$

where $\hat{y}$ is the predicted due date and the starred values are the solution values found from solving the equations above.

OLS Regression

The OLS regression formulation is the same as the quadratic programming model, namely objective (11) subject to constraints (9) and (10).

As discussed earlier, only models linear in the x's of equation (9) have been reported with success in the literature. Nonetheless, in this research we examine both the case of linear and nonlinear independent variables.

Now that the mathematical formulation for determining costs has been specified for each of the solution approaches to be considered in this chapter, the next section sets out a procedure for determining the relative performance of these techniques for different cost functions.

Procedure

This section of chapter 4 defines the procedure to be utilized in determining the relative performance of the neural network and the classical approaches in cost-based due-date prediction. The methodology employed is a statistical comparison of results obtained from predictions in a simulated shop. That is, a simulated shop is used to generate data indicating how long it takes jobs to complete. These data are then used to assess the prediction capabilities of the aforementioned techniques with different assumed cost functions.

Shop Simulation

The shop chosen for study is indicated in Figure 4-2. Jobs entering the shop are routed through its five workcenters in a sequential order. At each workcenter jobs are processed at one of two available machines, the determination of which is made for each job when it enters the shop.

As discussed earlier, internally-set due dates are the focus of the study, an order-release delay of zero (i.e., no delay or immediate release) is assumed when releasing jobs to the shop, and the shortest processing time (SPT) dispatching rule is utilized. Machine processing times are drawn from a negative exponential distribution with a mean of 1.8. Job interarrival times are also selected from the same distribution, but with a mean of 1.0.

The SLAM II simulation language (Pritsker, et al. [59]) was used to model this shop. Once the simulation had reached steady state 3,500 jobs were generated, "pushed" through the shop, and the data described below were collected for each job. These data

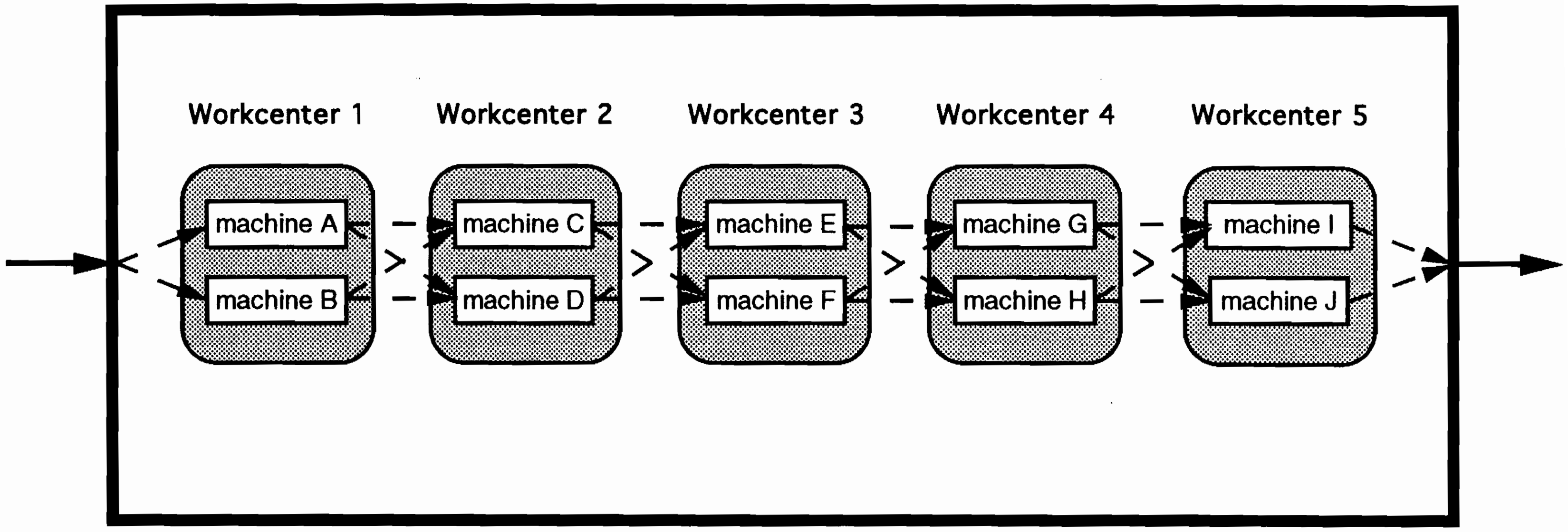


Figure 4-2: The shop used to test the methodologies

were used to train the neural network (determine its weights) and also to determine the coefficients in the mathematical programming and regression models. An additional 10,000 jobs were generated and processed by the shop for later use as test data. These data were used to compare statistically the performance of the different methodologies on the various cost functions studied, once all weights in the neural network had been determined by the 3,500 training data and all coefficients in the classical cases had been calculated from the same 3,500 data points. Each method's performance was evaluated by substituting the 10,000 data points into the respective prediction equations, from which the predicted due dates were calculated. The earliness and tardiness cost functions were then employed to determine the total cost penalty incurred by each prediction technique for missing due dates.

Both individual job characteristics and shop-status data were collected for the two sets (3,500 and 10,000) of data. In particular, for each job, eight general, job- and shop-characteristic data were obtained, as well as three workcenter-specific data at each of the five workcenters, resulting in 8+(3*5)=23 characteristics per job (see Table 4-1). In addition, the actual time of exit from the shop was observed. This value was assumed to be the "true" value of the due date, and hence was used as the desired due-date value. Except for the actual flow time, all data generated were collected when the job entered the shop. As a result, all shop-status data are reflective of the condition of the shop at the instant of job entry, as is desired to predict due dates.

Experimental Design

Although further practitioner-oriented research is needed to determine realistic cost functions, we suspect that earliness- and tardiness-cost functions differ in form from

Table 4-1: The 23 inputs for each job j

Input	Information
	General Shop Characteristics
1	number of operations required by job j
2	sum of processing times for job j
3	number of operations that must be done by the machines on job j's path to complete all jobs presently in the shop
4	sum of jobs presently in queues on job j's routing
5	maximum processing time job j requires on any machine
6	number of operations required to empty the shop of its current workload
7	sum of the expected waiting times at machines on job j's routing
8	total processing time of jobs in queues on job j's routing
	Workcenter-Specific Characteristics
9,...,13	processing time for operation 1,...,5
14,...,18	number of operations which must be done on the machine which does operation 1,...,5 on job j in order to complete all the jobs which are presently in the shop
19,...,23	number of jobs presently in queue on the machine which does operation 1,...,5 on job j

each other since the factors generating these costs differ fundamentally. For example, holding costs are often taken to increase linearly with time in the inventory literature, whereas tardiness costs include the cost of ill will, generally assumed to be nonlinear with time. Furthermore, it is likely that tardiness costs increase rapidly with increasing tardiness.

In this study we consider polynomial cost functions of degree three or less and create five combinations of these. To consider the case whereby earliness and tardiness cost functions differ in form, we evaluate scenarios that have asymmetric costs for earliness and tardiness, in the sense that they differ both in form and in magnitude. Further note that using first- and second-order polynomial costs allows us to evaluate cases that would be expected to be as favorable as possible for the conventional methods mathematical programming and regression, as those techniques implicitly assume such cost functions. Directly including the case whereby cost functions are of high polynomial degree is not so easily met, however, because of the difficulty in solving due-date predictions with such cost functions under the conventional techniques without using search heuristics. Nonetheless, we still include a third-order polynomial cost function, anticipating that it will not be as favorable, for example, as the second-order case for OLS regression. In fact, we expect that polynomials of even higher order would result in progressively poorer performance for the conventional techniques as the explicit cost functions differ more and more from the implied first or second order form. Note, however, that neural networks do not inherently have such limitations. In fact, neural networks can be used to model any earliness and tardiness cost functions that are differentiable, a potentially decided advantage.

Figure 4-3 shows the five cost functions considered. Each of the cost functions illustrated is a plot of cost versus error in prediction, and each curve goes through the origin, which indicates that no cost is incurred if the due date is exactly met. A positive error indicates tardiness, and a negative error earliness.

The five cost functions evaluated are

(1) Linear 1:1

$$C(e_1) = \begin{cases} e_1, & e_1 > 0 \\ -e_1, & e_1 < 0 \end{cases} \tag{12}$$

(2) Linear 1:10

$$C(e_1) = \begin{cases} 10\, e_1, & e_1 > 0 \\ -e_1, & e_1 < 0 \end{cases} \tag{13}$$

(3) Quadratic 1:1

$$C(e_1) = \frac{1}{2}\, e_1^2, \quad e_1 \neq 0 \tag{14}$$

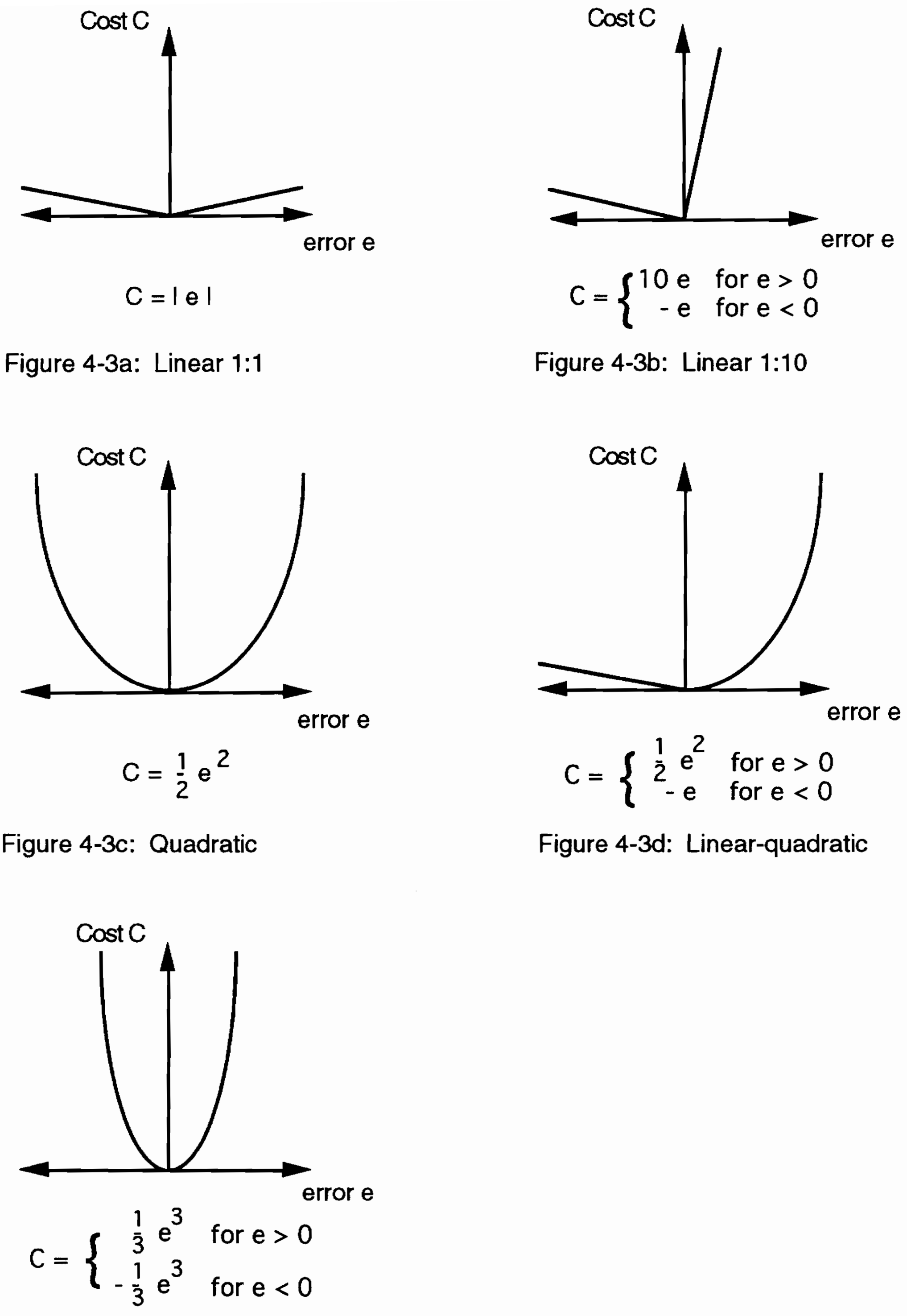
(4) Linear-Quadratic

$$C(e_1) = \begin{cases} \frac{1}{2} e_1^2, & e_1 > 0 \\ -e_1, & e_1 < 0 \end{cases} \tag{15}$$

(5) Cubic 1:1

$$C(e_1) = \begin{cases} \frac{1}{3} e_1^3, & e_1 > 0 \\ -\frac{1}{3} e_1^3, & e_1 < 0 \end{cases} \tag{16}$$

The experimental design used to evaluate the techniques on the different cost functions is shown in Table 4-2. As shown in that table, neural networks are used with all five cost



$$C = |e|$$

Figure 4-3a: Linear 1:1

$$C = \begin{cases} 10\,e & \text{for } e > 0 \\ -e & \text{for } e < 0 \end{cases}$$

Figure 4-3b: Linear 1:10

$$C = \frac{1}{2} e^2$$

Figure 4-3c: Quadratic

$$C = \begin{cases} \frac{1}{2} e^2 & \text{for } e > 0 \\ -e & \text{for } e < 0 \end{cases}$$

Figure 4-3d: Linear-quadratic

$$C = \begin{cases} \frac{1}{3} e^3 & \text{for } e > 0 \\ -\frac{1}{3} e^3 & \text{for } e < 0 \end{cases}$$

Figure 4-3e: Cubic

Figure 4-3: The five different cost functions studied

Table 4-2: The experimental design

Prediction Technique	Form of the Cost Function				
	Linear		Quadratic	Linear-Quadratic	Cubic
	1:1	1:10			
Neural Networks	x	x	x	x	x
Linear Programming	x	x			
Quadratic Programming			x	x	
OLS Regression	x	x	x	x	x

Note: x indicates a design point
blank indicates no design runs made

functions. Linear programming is used with the two linear cases; quadratic programming is evaluated on the two cost scenarios that have a quadratic component; and OLS regression is also tested across all five cases. Further elaboration of the regression case is necessary, however. As mentioned, the literature is practically void of research using regression that allows nonlinearity in the variables. We therefore selected six of the better performing linear rules according to Ragatz and Mabert's [62] comparative study, and explore each of these. We conjecture that practitioners use rules of thumb similar to these. The six rules chosen are

(1) RMR linear (the best linear regression based on response mapping procedures that identify the independent variables and estimate various functional rule equations),

(2) WIQ (a regression based on the job's total processing time and the total work in queues on its path),

(3) JIQ (a regression based on the job's total processing time and the total number of jobs in queues on the job's routing),

(4) TWK+NOP (a regression based on the job's total processing time and the number of operations on its path),

(5) NOP (a regression based only on the number of operations on the job's routing),

(6) TWK (a regression based only on the job's total processing time).

The results from chapter 3 suggested that allowing variables to be nonlinear significantly improves the ability of OLS regression to fit due-date data, although no cost-based comparison was made. In particular, we considered the case where the same variables chosen in the (linear) RMR case above are also allowed to have second order terms (including cross products). Because of our success with this "nonlinear" RMR model, we include it as well in our experimentai design for all five cost cases. By so doing, we hope to see whether the omission of nonlinear considerations in the literature and in practice is costly, and thus to determine whether further study is necessary in this area.

Neural Network Implementation

A backpropagation network with 23 input processing elements, 9 hidden layer PEs, and 1 output PE was utilized for this phase of the study, and is indicated in Figure 4-4. Each of the 23 input nodes was supplied with one of the 23 data characteristics generated for each job, with the one output node representing the due date. The bias node shown in that figure provides a constant (equal to one) input and acts much like an intercept term in a regression model; bias is universally included in backpropagation networks.

Generally, in a backpropagation network, all PEs in the input and output layers are fully connected to the hidden layer, and the number of PEs in the hidden layer is chosen by a rule of thumb such as "use half the sum of the input and output nodes," which in this case would have been 0.5*(23+1)=12. Only nine hidden layer nodes were used here, however, and the PEs were not fully connected because shop configuration information was utilized implicitly, thereby reducing complexity of the neural network, as will now be explained. As shown in Figure 4-4, eight of the input nodes representing general characteristics were fully connected to the first four hidden layer nodes, as is common practice. However, each of the other five hidden layer nodes was only connected to three of the input nodes. This is because there is no reason to expect activities at one operation to affect significantly those at another, given the randomness with which the jobs are routed through the shop. Therefore, hidden layer node 5, for example, was used to learn about the first operation (only), as its only three inputs were nodes 9 (the processing time for the first operation for this job), 14 (the anticipated total time to clear the system at this operation), and 19 (the current number of jobs in queue at this

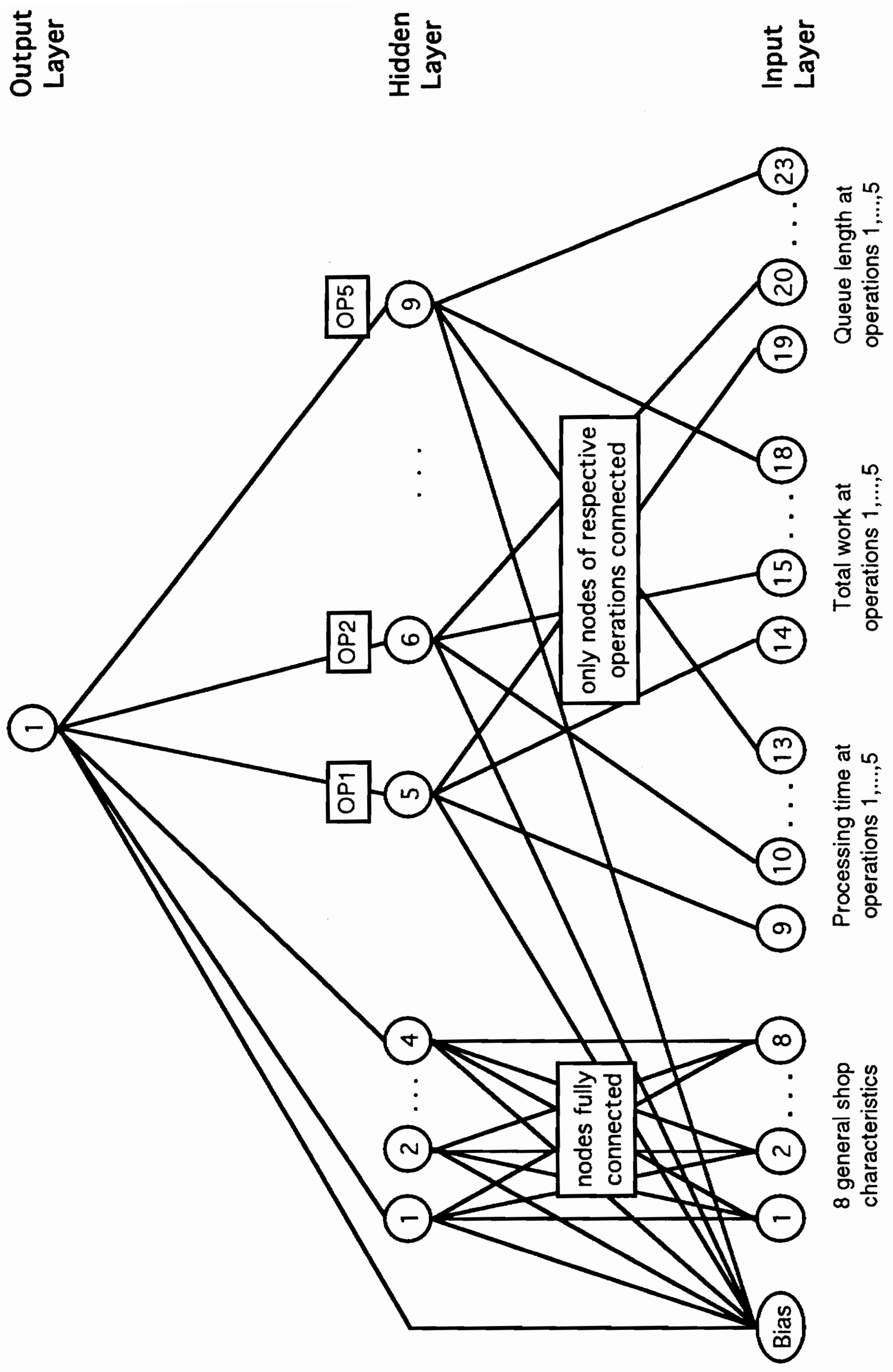

Output Layer
Hidden Layer
Input Layer
1
OP1
OP2
OP5
1
2
. . .
4
5
6
. . .
9
nodes fully connected
only nodes of respective operations connected
Bias
1
2
. . .
8
9
10
. . .
13
14
15
. . .
18
19
20
. . .
23
8 general shop characteristics
Processing time at operations 1,...,5
Total work at operations 1,...,5
Queue length at operations 1,...,5

Figure 4-4: The backpropagation network used for due-date prediction

operation). This particular neural network has also worked well in the due-date research we have conducted both in shops of varying structure and across different sample sizes of data used to fit the models as presented in the previous chapter 3, and hence we use it again here.

In the classical cases (involving regression and mathematical programming) to be described momentarily, 3,500 data points were generated to estimate coefficients. In the neural network case, these 3,500 points were used both to train the network and then to decide when to stop training.

Deciding when to stop training a neural network is not easy, since neural networks often overtrain (so called "grandmothering") in that they learn more than just the functional relationship underlying the data; they learn the peculiarities and vagaries of the particular data set used to train. As a result, in a solidly-designed study, a second data set is used to decide when to stop training. This is done by evaluating the performance of the trained (by the first data set) network progressively at different numbers of training cycles with the fresh (second) data set. The training then is stopped when the trained network produces best performance on the second data set.

We arbitrarily allocated 80% of the 3,500 data points (i.e., 2,800) to training, holding back the other 20% (700 points) to decide when to stop. The specific manner in which we did this is as follows: we first trained the neural network for 25 training cycles of all 2,800 data points (i.e., 25*2,800 random presentations to the network), then we froze the weights and applied the other 700 data points to the network to determine performance according to the cost criterion. This gave us our first stopping data cost point. We then retrained the network with 50 presentations of all 2,800 data points, froze the new weights and reapplied the 700 data points to again determine cost performance. This process was repeated until a cost minimum was found (and further increases in the number of presentations led to continued values in excess of that minimum because the network was grandmothering). Then the minimum point on the "curve" was chosen as the point to stop training, which for the cases considered in this chapter was always in the range of 150 to 225 training cycles. For each case, once the optimal training point was determined, we retrained the network with all 3,500 points (not wanting to waste any data) presented to the network a number of times equal to the optimal number of training cycles found.

The training, stopping, and recall procedure described above was implemented on an IBM 3090 (mainframe) computer, using extensive software written by the author in a third generation language.

Classical Methods

The implementation of the classical methods was more straightforward because commercially available computer packages were utilized. The MINOS (version 5.1) package [54] was used for both the linear and nonlinear programming tasks, whereas the SAS Institute's [70] OLS regression module was invoked for both the regression work and the practitioner cases. Again an IBM 3090 was used. For each of the classical methods the 3,500 data points were used to estimate the coefficients of the prediction line by running the respective software packages.

Statistical Testing

Once all the coefficients in the classical methods had been determined as described above and all the neural network weights had been set during the training phase, all coefficients and weights were frozen and tested using the additional 10,000 data points generated specifically for that purpose from the same shop. The same 10,000 points were presented to the classical methods as to the neural network; hence, a paired statistical test (the paired-t test) was suggested.

Recall that the 10,000 data points were generated from the SLAM II simulation model. It should be noted, however, that these jobs were not generated sequentially from the model. Rather, there was only a 2% chance that any simulation output produced would be chosen as one of our 10,000 data points. Stated differently, on average, every fiftieth job generated by the model was chosen to be in our 10,000 data point sample. This procedure was followed to reduce serial correlation between jobs. Nonetheless, to further insure that the independence and normality assumptions of the paired-t test were met, we used the "batch-means method" (see, for example, Law and Kelton 1982) and formed 10 batches of size 1,000 each. Further, recall that since only one population exists in the paired-t test (that of the differences), there is no need to test for homoskedasticity of variances.

Since we have had previous success with neural networks as due-date predictors in chapter 3, we formulated the one-sided hypothesis that the cost of any method other than neural networks taken alone would not exceed that obtained from the neural network, a hypothesis we hoped to reject with statistical significance. In particular, we tested:

$$H_0: \ \mu_D \leq 0$$

$$H_1: \ \mu_D > 0,$$

where D is the difference between the particular classical method's cost and that of the neural network. For all tests we set the confidence level at $\alpha=0.01$, using Bonferroni's inequality to control the experimentwise error rate of the multiple comparisons.

Results

In summary, the procedure described above was followed to generate 3,500 data points from the shop shown in Figure 4-2; the shop assumed immediate release and SPT dispatching. Those data points were used to estimate coefficients in the classical case and the PE weights in the neural network case. Then 10,000 test points were grouped into ten batches, and the paired-t test was used to test the hypothesis that the neural network did not perform better than its competition in each case considered while the experimentwise error rate was controlled by using Bonferroni's inequality. The cases considered are indicated by the x's in Table 4-2, where the experimental design is shown to consist of four techniques studied on five cost functions. Results from each of the five cost function cases are now presented.

<u>Linear 1:1 Cost Function</u>

In this scenario the linear cost function of equation (12) with equal penalties for both earliness and tardiness is employed. It is the cost function implicitly utilized with a mean absolute deviation (MAD) criterion. The MAD criterion is often used in the manufacturing literature, for example in Ragatz and Mabert [62], and also was used in chapter 3, although it is not at all certain that such a cost function often squares with reality.

The results from the batch-means comparison for the top two performing techniques are shown in Table 4-3. That table indicates that the neural network (NN) at $156,443 outperforms the LP approach ($157,514), but only slightly by about 0.7%. The results for the nonlinear RMR regression indicate an overall cost of about $173,000, which is significantly higher than either the NN or LP result. This is not a surprising

Table 4-3: Cost comparison for a linear 1:1 function

Batch Number	Cost of NN Prediction	Cost of LP Prediction	NN Cost Advantage
1	13,937	14,106	169
2	15,938	16,115	177
3	16,304	16,516	212
4	16,484	16,808	324
5	13,425	13,167	-258
6	17,326	17,557	231
7	14,504	14,597	93
8	14,970	15,122	152
9	16,355	16,483	128
10	17,200	17,043	-157
Sum	$156,443	$157,514	$1,071

result as regression implicitly assumes a quadratic cost function, and we are specifying a linear one. The comparison between only NN and LP indicates a p-value of less than 0.05, but since NN has been explicitly compared against LP and all OLS regression techniques in our test hypothesis, we use Bonferroni's inequality to control the experimentwise error rate. The result of this latter test indicates no significant cost difference.

Results (not shown in the table) from using the simpler rules found in the literature indicate costs ranging from $185,000 to $228,000, depending on the rule, thereby demonstrating the magnitude of additional costs incurred when the optimal technique is not used. Another way of viewing this result is presented in the first column of Table 4-4, which shows the relative performance of all methods compared to the best (where NN=1.00).

Linear 1:10 Cost Function

This scenario considers the cost function of equation (13), where the cost of being tardy is ten times that of being early, but both functions are linear in the magnitude of the error.

The batch-means comparison indicates that using LP is (approximately) 12% more costly than using neural networks, the p-value of this pairwise comparison being less than 0.0005. Moreover, the nonlinear RMR regression is a distant third, 54% higher than the NN solution. Using Bonferroni's inequality in this case indicates the neural network is preferred over any other methodology at the experimentwise significance level of α=0.01.

Table 4-4: Relative performance of the different techniques on the five cost functions

	Form of the Cost Function				
Prediction Technique	Linear 1:1	Linear 1:10	Quadratic	Linear-Quadratic	Cubic
Neural Networks	1.00	1.00	1.00	1.00	1.00
Linear Programming	1.01	1.12			
Quadratic Programming			1.07	1.27	
Nonlinear RMR	1.10	1.54	1.02	2.89	1.16
Simpler Regression-Based Rules					
RMR (linear)	1.18	1.63	1.08	3.14	1.25
WIQ	1.24	1.59	1.17	3.54	1.38
JIQ	1.24	1.58	1.17	3.55	1.39
TWK+NOP	1.20	1.69	1.15	3.45	1.35
NOP	1.45	2.07	1.43	4.43	1.67
TWK	1.22	1.59	1.17	3.55	1.39

The details of the NN versus LP case demonstrate the superior prediction performance of the neural network, which is able to reduce the number of costly, tardy jobs. For the 10,000 recall data points, NN predicted:

# early	9,074	early cost (in $1,000):	$ 285
# tardy	926	tardy cost (in $1,000):	$ 357
totals	10,000		$ 642

whereas the LP determined:

# early	8,997	early cost (in $1,000):	$ 291
# tardy	1,003	tardy cost (in $1,000):	$ 430
totals	10,000		$ 721

Further note that although the neural network had 77 more early jobs, it was able to fit early jobs better than the LP could, thereby having fewer very early jobs and thus incurring a smaller earliness cost.

Finally, the six simpler regression-based approaches yield results similar to the nonlinear RMR regression case (see Table 4-4), which again indicates costly due-date prediction consequences.

Quadratic Cost Function

In this case a (symmetric on both sides) quadratic cost function is assumed, namely that of equation (14). The NN results in costs about 2% less than those from the best OLS-regression technique, which again was the nonlinear RMR. The costs incurred by using quadratic programming run approximately 7% higher than those from the neural network predictions. Comparing just quadratic programming and nonlinear RMR regression clearly shows the benefit of also including the independent variables in a nonlinear fashion, which is the major difference between these two methods. Again, note the relatively poor performance of the simpler regression-based rules for this cost function in Table 4-4. Overall, the neural network was again the best technique, resulting in a p-value of approximately 0.1 for the two-way comparison with the nonlinear RMR, which was insignificant in the comparison against all other techniques.

Linear-Quadratic Cost Function

As argued above, it is conceivable and perhaps even likely that the form of the earliness cost function would differ from the tardiness cost function. In this scenario we consider the case where earliness is considered linear with the amount of earliness, and tardiness is deemed quadratic with the tardiness involved (see equation (15)).

Results indicate that the QP as the best conventional method is about 27% more costly than NN prediction. The paired-t test of the NN against all other methods is statistically significant at $\alpha=0.01$.

Finally, it should be noted that the best regression result (the nonlinear RMR model) is almost 2.9 times more costly than the NN solution. Results from the other six regression-based rules are worse than that, ranging on up (in the worst case) to over 4.4 times the cost of the NN.

Cubic Cost Function

When a cubic cost function (equation (16)) is used, the nonlinear RMR regression solution incurs 16% higher costs compared to using NN due-date prediction. The fact that NN outperform OLS is not surprising, as regression assumes a quadratic cost function, whereas neural networks use the actual cost function; but the magnitude of the cost savings is drastic. The other six regression rules give costs that are again worse, demonstrating the costliness of proceeding with just those assignment rules currently listed in the literature. Moreover, note that in this scenario it is impossible to even attempt quadratic programming per se, as the cubic form of the cost function precludes non-heuristic techniques. In this case again, the neural network shows statistically significant better performance than all other methods considered.

Conclusions and Future Work

This chapter has presented a methodology for explicitly considering costs in the due-date prediction problem. It was pointed out that practitioners and researchers are implicitly assuming *some* cost function, quadratic and symmetric if OLS regression is used to set the due dates, or linear if linear programming is used.

With respect to a best approach for due-date prediction, four techniques were evaluated, but only for a single shop. These techniques consisted of neural networks, OLS regression, and mathematical programming (both linear and quadratic). It was found for the cases considered that: (1) implicitly ignoring cost-based predictions can be very costly; (2) simpler regression-based rules cited in the literature are very poor cost performers; and (3) if the form of the earliness cost function differs from that of the tardiness cost function, neural networks are statistically superior performers; using other methodologies can again be very costly. Moreover, neural networks can be used for essentially any differentiable cost function desired, whereas the other methodologies are significantly more restricted.

The fact that the neural network performed as well as it did on all the linear, quadratic, cubic, and combined cost functions suggests a certain robustness with that technique across cost functions. This would be important to a firm that is not sure of the functional form of its costs or one that suspects its costs may change over time.

It was interesting to us to note that the neural network achieved its particular advantage during the recall phase of the process, the phase of interest in practice. That is, during the training phase (on the 3,500 data points), the neural network obtained better fits than its competitors, but not by nearly as wide a margin as during the recall phase. Hence, at least for the cases considered here, we conclude that the particular advantage of neural networks is in their power to generalize to data they have not seen.

With respect to the results, it should be noted that the purpose of the experiment was first to show how cost-based considerations can be directly included in studies; second to show how an artificial intelligence technique, neural networks, can be successfully used in this scenario; and third, to suggest that omitting nonlinear terms in regression-based due-date settings is costly. One should therefore be careful in interpreting our results. Only one particular shop has been examined, and neither the neural network nor the regression techniques were optimized. The point rather was to illustrate the need for explicit cost considerations and to demonstrate the wisdom of the inclusion of nonlinear approaches. Recall that the best regression technique in all cases considered assumed nonlinear independent variables while the linear counterparts were significantly more costly.

Although neural networks when compared costwise to their competitors are better in most cases and, in fact, are no worse in any of the cases considered here, they have not yet penetrated many business firms. Even so, we do not believe this approach is difficult to implement. It has been demonstrated in the second section of this chapter how to include any earliness and tardiness functions that are differentiable. Thus, theoretically, a firm could use any such cost function it experiences by using equation (7). Conversely, classical approaches cannot be modified so easily. For instance, in a case not reported above, MINOS [54] was used to solve a cubic cost function. As a closed-form solution technique to this problem is not known, MINOS conducted a search, and (initially) got "stuck" at a local minimum after 29 minutes of CPU time. We believe this kind of problem would be difficult for practitioners to deal with. Regression is probably the most familiar technique, which is a decided advantage, but it performs poorly in all cases other than the symmetric quadratic scenario. Moreover, the LP and QP formulations were somewhat large (with roughly 3,500 constraints). Because of size, they could not be run on the versions of LINDO available at our workplace. They also took considerable solution time (28 and 51 minutes of CPU time respectively on an IBM 3090 mainframe computer using the MINOS version 5.1 package).

This research suggests that future work is needed in three areas. In the first area, further study of practitioner issues should be made. This includes examining other shops and determining and evaluating "real-world" cost functions for both earliness and tardiness. In the second, additional study of regression is needed allowing nonlinearity in its variables. And in the third area, the one that motivated the following chapter 5, more theoretical work integrating the three stages of control, namely due-date prediction, order-release, and dispatching, on a cost basis appears promising and worthwhile.

Chapter 5

INTERACTIVE DUE-DATE PREDICTION WITH ASYMMETRIC COST FUNCTIONS

Introduction

As first suggested by Baker in 1984 [5] and reiterated by Ragatz and Mabert in 1988 [61], the production control system can be viewed as consisting of three sequential stages: the order-promising/master-scheduling stage, the order-release portion, and the shop-floor stage. In stage one, order-promising/master-scheduling, jobs arrive from the customer and are assigned a due date. If due dates are specified by the production department, they are referred to as "internally-set due dates;" conversely, if due dates are set by an order-entry marketing department in agreement with the customer, they are called "externally-imposed due dates" (see, for example, [62]). This distinction emphasizes the control, or lack thereof, of the production department in its effort to meet its performance objectives. In the second stage, order release, work is sent -- possibly after some intentional delay -- to the shop floor. In the final stage the work itself is performed; during this stage the sequence of work at various constrained workcenters must be determined. This decision is referred to as "dispatching."

Because of the complexity of the production control system, the literature has typically not examined all three stages in toto. Likewise, in this chapter we concentrate on the first stage of the production control system for a shop, namely order promising. However, this research also explores the value of interactively combining the first and third stages in such a way that the prediction mechanism in the first stage is continually updated based on how accurate these predictions appear to be after the third stage. Two different interaction schemes are studied, as will be discussed later; one is based upon OLS-regression due-date predictions, and the other is grounded in a neural-network approach. In both cases, due dates are predicted in stage one; then jobs are sequenced through the shop using those predictions and so-called "due-date-based" dispatching rules. We now examine these rules.

With due-date-based dispatching rules, jobs take their positions in queues at workcenters according to a job's due date. For example, with the MINSLK rule, jobs having the least slack time are placed at the front of the line, and will therefore be processed first. Similarly, with the earliest due date (EDD) rule, jobs are ranked in queues according to when they are due. Due-date-based rules are important because they are often less costly than other rules, such as first-in, first-out (FIFO). For example, Ragatz and Mabert [62] found for cases they studied that the MINSLK rule performed better than the non-due-date-based rules, shortest processing time (SPT) and first-come, first-served (FCFS), according to mean-absolute-deviation and standard-deviation-of-lateness criteria for several due-date prediction techniques. However, prediction of due dates in a due-date-based dispatching scheme can be much more difficult than when other approaches are used. The reason for this is readily apparent. With a simple dispatching rule such as FIFO, progress through the shop does not depend on the due date itself; rather jobs join and leave queues in the order they arrive. Alternatively, with a due-date-based rule, jobs at workcenters may get "bumped," regardless of how long they have been in line, by newly-arriving, imminently-due jobs. In this latter case, the flow time through the shop for a given job, and hence its due date, may depend on jobs which customers have not even brought to the firm yet.

As mentioned, the economies of shop floor control warrant study of due-date-based predictions. But although this topic is important, almost all such studies use what we term to be non-interactive prediction mechanisms. That is, a forecasting scheme is established *a priori*, and predictions are made using that approach regardless of how well it is doing, how congested the shop is, etc. We believe that an interactive prediction approach holds promise in the sense that adjustments can be made to the prediction mechanism itself as exogenous and endogenous factors dictate. Nonetheless, we are aware of only one study to date (see Smith and Gee [75]) using an approach similar to such an interactive methodology. In our interactive approaches to be described below, prediction parameters are continuously updated. That is, every time a job leaves the shop, we update our prediction scheme. Smith and Gee used a batch, iterative approach in that they made a full simulation run of many jobs, stopped the simulation, collected data, and then fitted a regression (prediction) curve through the data. They then re-ran the simulation for many more jobs, with the new prediction parameters, stopped it, collected data again, re-fitted the regression, etc. They iterated for ten such batches of simulation runs. Smith and Gee reported success with their iterative procedure over a non-iterative approach for several performance criteria.

The issue of which performance criteria to use to evaluate these various interactive and non-interactive predictions raises an important consideration. "Traditionally" the literature has used performance criteria such as mean absolute deviation (MAD) and standard deviation of lateness (SDL) to evaluate proposed prediction rules. (See, for example, Ragatz and Mabert [62], or chapter 3 of this dissertation). But these types of criteria have an inherent difficulty: they implicitly assume symmetry in the cost function. That is, they assume that the cost or penalty for an early prediction is the same as the penalty for an equally tardy one. MAD tacitly assumes a linear, symmetric cost function; SDL makes the assumption of a quadratic, symmetric cost function. This, as we point out in chapter 4, is certainly not universally true in shops. In fact, we postulate that many firms actually have asymmetric cost functions, with linear earliness costs of holding, and higher-order tardiness costs, representing the cost of ill will. Therefore, monetary costs are again, as in the previous chapter, used as the performance criterion, thereby allowing for asymmetric early and tardy cost functions. The only stipulation required is that both the earliness- and the tardiness- cost functions must be differentiable.

In chapter 3 we showed that neural networks outperformed nonlinear, linear, and practitioner regression-based rules in predicting due dates in an SPT (non-due-date-based) shop using SDL and MAD criteria. In chapter 4 we demonstrated that neural networks never did worse than regression and mathematical programming approaches over a wide variety of cost functions. We also found that for asymmetric costs, neural networks outperformed the regression, mathematical programming, and practitioner approaches in a statistically significant manner. Again, however, this was under SPT conditions. Since Ragatz and Mabert [62], among others, have found economic justification for due-date-based dispatching rules as discussed above, we therefore extend that work to the case with interactive prediction, allowing for asymmetric cost functions, considering both an interactive, neural network approach as well as an interactive OLS-regression method.

The methodology employed here is to generate data from a simulated shop. A neural network predictor as well as a regression forecaster are embedded within the shop. Statistical tests are performed on the two cases under a range of (symmetric and asymmetric) cost functions and are compared to a non-interactive baseline scenario.

Comparisons with non-due-date-based results are also made, although these comparisons are not the focus of this work.

The contribution of this chapter's research is three-fold. First, due-date prediction and shop-floor dispatching (stages 1 and 3 of the production control system) are integrated. Second, a methodology of interactive training procedures for neural networks and for regression-based due-date prediction techniques is developed. Third, this integrated methodology is compared on a cost basis to examine three questions: is there an advantage to interactive training; if limited to the regression world, is there reason to use models nonlinear in their variables, a situation currently not practiced; and, what is the best prediction technique for due-date forecasting? It is found that the answer to the first two questions is affirmative, and that the resolution of the third question depends on whether the earliness/tardiness cost function is symmetric.

The plan of presentation for this chapter is as follows. In the next section we present two approaches to interactive training, which is the focus of this research, and in the third section the experimental design is discussed. The fourth section details the four-step procedure used in the study. Results of the experiments are furnished in the following section, and the chapter finishes with a statement of conclusions and a discussion of future work.

Two Approaches To Interactive Training

In this section two interactive training approaches to the due-date prediction problem are presented. As indicated, each of these methods will entail continuous updating of parameters as jobs leave the shop. The first training mechanism discussed is the AI-based, neural network approach, and the second is the OLS-regression formulation.

Certain fundamental portions of the prediction mechanism are the same for both approaches, as is illustrated in Figure 5-1. As suggested by that figure, jobs are brought by customers to the shop (see step 1, as indicated by the encircled "1"). Since due dates are internally set, the production department sets the projected time of system exit, as symbolized in Figure 5-1, step 2, by the due date attached in a circle to the job. This date is set after consultation with the "PREDICTOR," which itself obtains information on job characteristics and current shop conditions, as indicated by the in- and out-arrows at the left and bottom of that box. Since we are not studying order-release mechanisms (stage 2 of a production control system), immediate release of this job to the shop floor occurs. This job then proceeds to the machine of the first workcenter at which it must be processed, where it finds its place in the queue according to the due-date-based rule (say EDD) in effect in the shop. The job continues to move through the shop from workcenter to workcenter in the sequence necessary to finish all required processing. Meanwhile, further jobs are arriving and are assigned due dates, and yet other jobs are circulating on the shop floor. Finally (step 3), the job of interest completes and exits the shop floor. Immediately upon exit, the job's due date is compared with the actual date (see the calendar in Figure 5-1), and this datum is fed to the PREDICTOR, where changes to the prediction mechanism itself can be made. In this manner, the prediction of due dates for subsequent jobs can hopefully be improved.

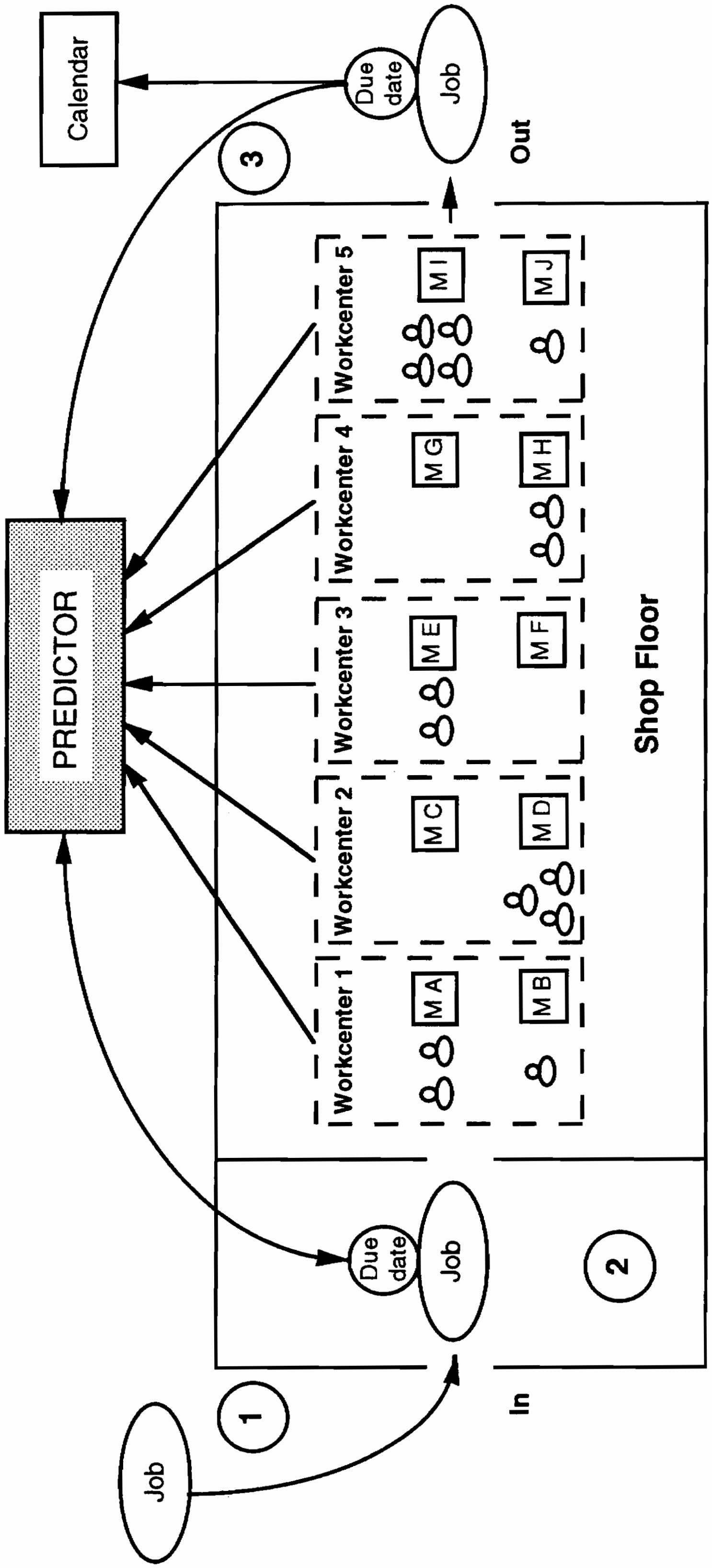


Figure 5-1: The general due-date prediction mechanism

In actuality, the PREDICTOR sets its due dates using a "rolling horizon" of the last 100 jobs to complete (step 3 above). The idea is that, in this manner, not just the last job, but the most recent shop performance as a whole will be included in current forecasts. Therefore, as any job finishes, its actual due-date performance and its predicted due date are made available to the PREDICTOR, which also utilizes the same data for the previous 99 jobs.

Note that when the very first job enters the shop (at point 1 in Figure 5-1), the PREDICTOR has no historical data on which to base its prediction. In fact, no such data will exist until jobs start exiting the shop. Consequently, the PREDICTOR must be pre-trained on some data set, regardless of which of the two cases of training is involved. Hopefully, the pre-training data set will encompass data somewhat like the actual jobs the shop will see, and the PREDICTOR will not "behave too badly" until it gets to see some "real" data. In this research, for both the neural network and the regression cases, the pre-training data consists of 500 jobs generated under non-interactive conditions. In particular, jobs are produced for this purpose in a shop with EDD dispatching, where the due dates are set by a simple rule of thumb. The heuristic is: the predicted due date equals a constant times the sum of the processing times, where the constant is determined in pilot studies, rather than by an interactive procedure. These pre-training due-date data are then used to train both the neural network and the regression predictors in their own unique manners.

We now discuss those features of the PREDICTOR that are specific to each of the two approaches developed in this research.

Neural Network Interactive Training

As mentioned, PREDICTOR behavior is based upon a rolling horizon of the last 100 jobs to complete. The particular way in which this occurs in the neural network interactive training case is as follows. With reference to Figure 5-1, once any job reaches step 3 (i.e., it has finished), the actual and predicted due dates for this job are passed on to the PREDICTOR. There this job and the previous 99 are presented randomly to the neural network, where delta rule learning is used to update the weights of the standard backpropagation network. Thus, any future job arriving at the shop (step 1) will have its due date set by a forward pass through the neural network using the updated weights; these weights will again be updated when the next job exits the shop.

The previous discussion centering around Figure 5-1 and outlining the interactive training procedure glossed over one critical fact: there is a time lag between the moment a job's due date is predicted upon shop entry and the instant when the job leaves the shop. This is particularly consequential in the neural network case for the following reason. When a job arrives, the neural network is consulted for a due-date prediction; according to delta rule learning, this prediction is made with a forward pass through the network. The problem is that the backward pass through the network cannot be made at this instant, because no value of the actual completion date is known until the job leaves the shop. Hence, no error can be calculated, and no backward pass with concomitant weight changes made can occur at this time. Once the job leaves the shop, then an error can be calculated, but the problem then is to decide which of two possible error calculations to use, neither of which is totally desirable. One possibility is to calculate the error as the difference between the actual flow time and the initial due-date prediction made at job entry. But in this case the neural network weights have been changed by intervening jobs since the initial due-date prediction was made. Thus, (back)propagating the error through the network results in updating a set of weights no longer responsible for that error. The second possibility is to determine the error as the difference between the actual flow and a due date that would be predicted by the current neural network weights at the moment of job exit, but the drawback here is that the actual flow time is a result of the initial due-date prediction based on the earlier (job-entry) set of weights. In this case, an error would be (back)propagated through the network with a corresponding set of weights, but, as stated, these are not the weights that forecast the due date in the first place and subsequently determined the job's progress through the shop. Neither of these two possibilities is completely satisfying; each has an inherent flaw. We therefore performed every experiment reported in this chapter twice, once for each possibility above. Results were mixed, with neither procedure dominating the other. Consequently, we arbitrarily selected the second case for presentation. In the sequel, further mention of the first possibility is omitted.

OLS-Regression Interactive Training

The regression interactive training case does not have the difficulty above associated with the neural network. This is because there are not two operations in regression that become separated with a time lag by the interactive procedure.

A regression model was used to fit a curve through the last 100 jobs, once any job completed (reaching step 3). Variables were included in the regression using SAS [70] routines based upon the C_p statistic and the level of the mean squared error (see, for example, Myers [55]). Once the regression parameters were estimated, current due-date predictions were made using those estimates until another job completed, at which point a new regression run with the latest set of points was undertaken.

Experimental Design

The study conducted in this chapter centers on three separate experiments, each evaluated over four different cost scenarios. The four cost scenarios are indicated in Figure 5-2. Each of the cost functions illustrated is a plot of cost versus error in prediction, and each curve goes through the origin, which indicates that no cost is incurred if the data is exactly met. A positive error indicates tardiness, and a negative error earliness. In short, the experiments examined:

(1) Interactive training versus non-interactive training.

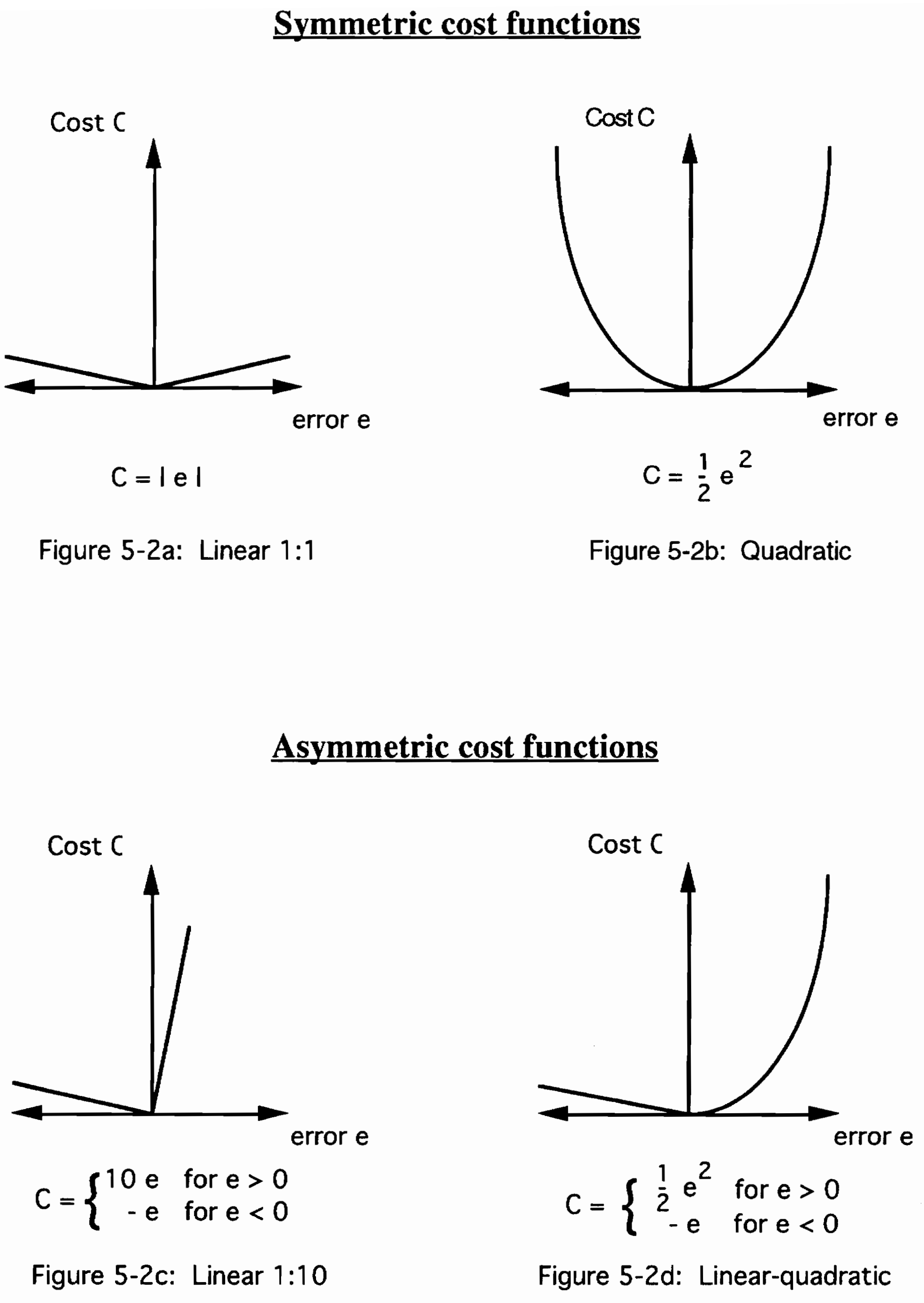
This experiment was conducted for five separate due-date prediction techniques: neural networks, regression (both linear and nonlinear-in-variables cases), and two practitioner-based approaches.

(2) Linear versus nonlinear (in variables) regression.

As the literature barely deals with the case of regression with nonlinear variables (for exceptions, see Smith and Gee [75] and the two previous chapters of this work), even in non-interactive settings, we examine the cost advantage of including nonlinear terms in regression models. In particular, we compare the better (non-interactive or interactive) linear regression model with the better nonlinear model.

(3) The better neural network case (i.e., interactive or non-interactive) versus the best regression case (interactive or non-interactive, linear or nonlinear).

The purpose with this test was to see whether the inclusion of neural networks as a prediction technique is worthwhile.



$C = | e |$

Figure 5-2a: Linear 1:1

$C = \frac{1}{2} e^2$

Figure 5-2b: Quadratic

$C = \begin{cases} 10\,e & \text{for } e > 0 \\ -\,e & \text{for } e < 0 \end{cases}$

Figure 5-2c: Linear 1:10

$C = \begin{cases} \frac{1}{2} e^2 & \text{for } e > 0 \\ -\,e & \text{for } e < 0 \end{cases}$

Figure 5-2d: Linear-quadratic

Figure 5-2: The four cost scenarios studied

In none of the three experiments do we have an *a priori* notion of which case would be preferred; therefore, for each experiment, we specify a two-sided hypothesis. Since the appropriate statistical tests are more readily apparent after the methodological procedure is explained, discussion of these will be delayed until that time. We do note at this point, however, that since the last two experiments are based upon the results of the first, so-called *a posteriori* (or *post hoc*) tests will be necessary.

Methodological Procedure

To determine whether interactive training is beneficial, and to decide whether neural networks are preferred to OLS-regression predictions, a four-step procedure was implemented. In the first step, a SLAM II [59] simulation model of a shop was written and validated. Moreover, neural network and regression PREDICTORS, as described above, were coded, and then imbedded in the SLAM II model. In the second step, pre-training of the two methods was conducted. In the following third step, ten separate simulation runs of 2,300 jobs each for all scenarios were generated (after a lengthy warm-up period). These jobs were then "pushed" through the shop and their earliness and tardiness calculated. In the fourth and final step, these data were used to compare statistically the performance of interactive training and the various techniques over symmetric and asymmetric cost functions. This procedure is now explained in more detail. All steps of the procedure were implemented on an IBM 3084 (mainframe) computer, using extensive software written by the authors.

Step 1: The Shop Simulation

The shop chosen for study is identical to the one used in the two previous chapters and is, again, indicated in Figure 5-3. Jobs entering the shop were routed through the five workcenters in a sequential order. At each workcenter the jobs were processed at one of two available machines, the determination of which was made for each job when it entered the shop.

Machine processing times were drawn from a negative exponential distribution with a mean of 1.8. Job interarrival times were also selected from a negative exponential distribution, but with a mean of 1.0. In addition, an order-release delay of zero (i.e., no delay) was assumed when releasing jobs to the shop. The earliest-due-date (EDD) dispatching rule was used in all cases discussed in this chapter of the dissertation.

Step 2: PREDICTOR Pre-Training

As mentioned, both the neural network and regression PREDICTORS were pre-trained on 500 data points generated under non-interactive conditions with EDD dispatching. This was necessary so that predictions could be made on the first jobs to enter the shop; only when jobs left the shop and errors were calculated could "real" data be incorporated into the PREDICTORS. Each data point consisted of the same information that was used once pre-training was over, i.e., twenty-one characteristics, as will be explained shortly. These characteristics include six general shop conditions at the time of job entry as well as fifteen workcenter-specific attributes. The particulars for the pre-training of each technique are now discussed.

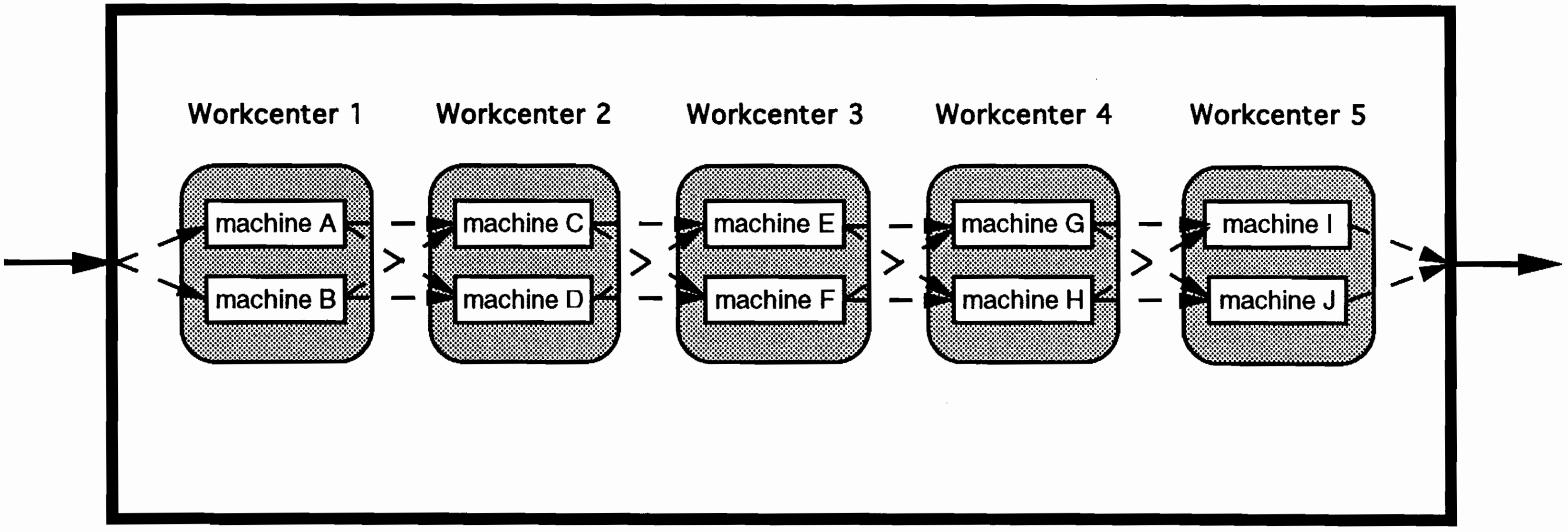


Figure 5-3: The shop used to test interactive training

Step 2a: Neural Network Pre-training

The neural network that must be pre-trained consists of three layers. The input layer has 21 nodes (one for each variable -- discussed in the next step), and the output layer has one output node (for the due-date prediction). There are nine middle layer neurodes.

Neural networks are trained with the three-step procedure (see Hinton and Touretzky [33]): initial training, with one data set; recall with a second data set; and then training for the number of cycles determined in the second step over both data sets. In the pre-training reported in this chapter, 400 of the 500 pre-training data points were presented randomly to the neural network for 25 complete times (i.e., 400*25 random presentations were made), thereby determining a set of neural net weights, which were then frozen. Then the other 100 data points, which the network had never seen before, were randomly presented to the frozen weights, and the performance of the network on the new points was noted. Then the weights were "unfrozen" and an additional 25 complete, random presentations of the original 400 data points were made, and the (new) weights re-frozen. Then the performance of the (frozen) network was again evaluated on the "new" data, namely the 100 data points on which it had not been trained. Again, performance was noted. This process continued until the ideal number of cycles (i.e., complete sets of presentations) was determined according to performance noted above. On the data set used here, between 125 and 275 cycles was optimal, depending on the cost function. Finally, the entire set of 500 data points was then presented randomly "from scratch" for the optimal number of cycles and the weights determined. This constituted the neural network pre-training.

Step 2b: OLS Regression Pre-training

As described earlier, the 500 data points were used to fit a regression curve for pre-training using SAS routines [70]. In one case variables were restricted to being linear; in the other case variables were permitted to be nonlinear as well as linear.

In the linear case, four variables were included: the sum of the job's processing times at all workcenters, the total work currently in all shop queues this job must visit, and the work at each of the last two workcenters taken separately. In the case where nonlinear variables were allowed to be included, twelve terms were present in the model, three of which were linear, and the rest of which were either squared or cross-product terms.

Step 3: Data Generation for Shop Testing

Since the literature suggests that both job and shop status data are helpful in due-date estimation (see e.g., Cheng and Gupta [14], Eilon and Chowdhury [20], Weeks [79], and chapter 3 of this work), we provided both categories of information to our PREDICTORS. That is, both individual-job-characteristic data and shop-status data were generated for each of the 500 pre-training jobs and the 2,300 actual-study jobs produced in this step of the methodology. In particular, for each job, six general, shop-characteristic data relevant to that job were produced as well as three workcenter-specific data at each of the five workcenters, resulting in 6+(3*5)=21 characteristics per job (see Table 5-1). All data generated were collected when the job entered the shop. As a result, all shop-status data are reflective of the condition of the shop at the instant of job entry, as is desired.

Table 5-1: The 21 inputs for each job j

Input	Information
	General Shop Characteristics
1	sum of processing times for job j
2	number of operations that must be done by the machines on job j's path to complete all jobs presently in the shop
3	sum of jobs presently in queues on job j's routing
4	maximum processing time job j requires on any machine
5	number of operations required to empty the shop of its current workload
6	total processing time of jobs in queues on job j's routing
	Workcenter-Specific Characteristics
7,...,11	processing time for operation 1,...,5
12,...,16	number of operations which must be done on the machine which does operation 1,...,5 on job j in order to complete all the jobs which are presently in the shop
17,...,21	number of jobs presently in queue on the machine which does operation 1,...,5 on job j

It should be noted that the jobs included in both the training and the experiment data sets were generated only after a significant warm-up period had transpired, thereby allowing shop transient effects to be eliminated. Also, in order to reduce serial correlation, which would render statistical tests more difficult, only every fiftieth job on average was included in either of the data samples. Stated differently, there was only a 2% chance that any job generated would be incorporated in either the pre-training or the actual-study data sets.

As has been discussed, the two data sets generated above were presented to the PREDICTORS in different manners. The pre-training dataset of size 500 jobs was presented to each of the neural network and regression predictors in one batch each. In the neural network case, the network was trained on the 500 data points. Then the neural network weights determined were used as the beginning weights in the study period for the neural PREDICTOR. In the regression case, the regression model was fitted to the same 500 data points, and the regression parameter estimates were then used as starting values for the regression PREDICTOR. The actual-study jobs, however, were not generated as a batch. In fact, each data point was produced at the moment it was to be presented to the shop. When a job entered the shop, it was assigned a due date by the appropriate PREDICTOR, utilizing the twenty-one specific job and shop-characteristic values determined at that instant. The job then moved through the shop, traveling from workcenter to workcenter and being placed in queues according to the EDD dispatching rule; that is, jobs were ordered at each machine dynamically from earliest to latest due date. When any job finished, its completion date was compared to its due date, and a cost (of earliness or lateness) was determined. At this point, the cost due to the error was passed on to the PREDICTOR, which could then adjust its subsequent predictions. This process was repeated for a total of 10 independent simulation runs of 2,300 jobs each, for each of the scenarios described in the sequel.

For this chapter's experiments, all data were generated from a hypothetical shop using a massive SLAM II [59] simulation program with neural network and regression PREDICTORS. The interactive network predictions along with the frequent neural network weight updates were performed by subroutines written and then imbedded into the simulation model by the author. Similarly, in the interactive regression case, every time a job finished, an IMSL subroutine [36] was called to update the regression coefficients that were used for subsequent predictions.

Step 3a: Neural Network Due-Date Setting

The basic functioning of the neural network predictor has been described above and will not be repeated here. As mentioned, the neural network code was imbedded in the simulation model. The network was pre-trained on 500 non-interactive data points, and once the study began, the network weights were updated after each job completed.

The neural network used in this chapter as the PREDICTOR is the same network used for pre-training. A backpropagation network with 21 input processing elements, 9 hidden layer PEs, and 1 output PE was utilized for this phase of the study, and is indicated in Figure 5-4. Each of the 21 input nodes was supplied with one of the 21 data characteristics generated for each job, with the one output node representing the due date. The bias node shown in that figure provides a constant (equal to one) input and acts much like an intercept term in a regression model; bias is universally included in backpropagation networks.

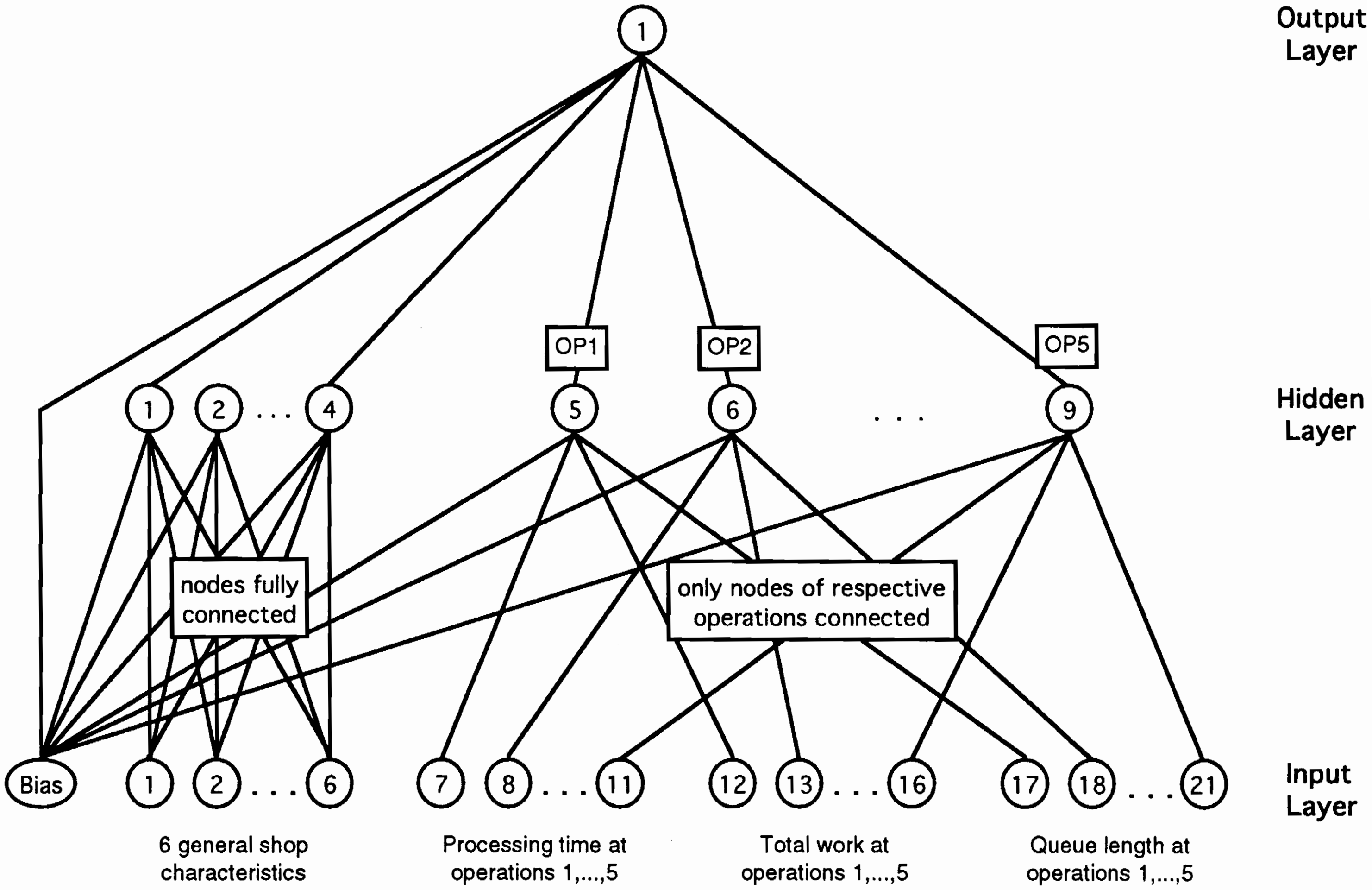

Output Layer
1
OP1
OP2
OP5
Hidden Layer
1
2
. . .
4
5
6
. . .
9
nodes fully connected
only nodes of respective operations connected
Input Layer
Bias
1
2
. . .
6
7
8
. . .
11
12
13
. . .
16
17
18
. . .
21
6 general shop characteristics
Processing time at operations 1,...,5
Total work at operations 1,...,5
Queue length at operations 1,...,5

Figure 5-4: The backpropagation network used for due-date prediction

Generally, in a backpropagation network, all PEs in the input and output layers are fully connected to the hidden layer, and the number of PEs in the hidden layer is chosen by a rule of thumb such as "use half the sum of the input and output nodes," which in this case would have been 0.5*(21+1)=11. Only nine hidden layer nodes were used here, however, and the PEs were not fully connected because shop configuration information was utilized implicitly, thereby reducing complexity of the neural network, as will now be explained. As shown in Figure 5-4, six of the input nodes representing general characteristics were fully connected to the first four hidden layer nodes, as is common practice. However, each of the other five hidden layer nodes was only connected to three of the input nodes. This is because there is no reason to expect activities at one operation to affect significantly those at another. Therefore, hidden layer node 5, for example, was used to learn about the first operation (only), as its only three inputs were nodes 7 (the processing time for the first operation for this job), 12 (the anticipated total time to clear the system at this operation), and 17 (the current number of jobs in queue at this operation). A very similar neural network has also worked well in the due-date research presented in chapters 3 and 4 both in shops of varying structure and across different sample sizes of data used to fit the models.

Step 3b: OLS Regression Due-Date Setting

The regression models used to fit the 500 pre-training data points have already been described; a four-variable model was fitted in the linear case, and a twelve-variable model was found to be preferred for the case with nonlinear variables included in the model.

For the actual study with the rolling horizon of 100 data points, the same regression model was utilized in the linear case as in pre-training. However, blindly using the pre-training 12-term nonlinear model would result in overfitting the data; 100 points are simply not sufficient to fit this model. Therefore, rather than exclude the nonlinear variable model altogether from consideration, we increased the "rolling horizon" size from 100 to 500 data points for the nonlinear case only, which (as we know from the pre-training case) is of sufficient size to fit the model. As a result, the same (12-term) variable model was used in the actual experiments as in the pre-training case. Note that increasing the nonlinear rolling horizon size from 100 to 500 has the advantage of allowing regression nonlinear in its variables to be evaluated, but suffers from the drawback that by having a longer memory, the 500-point model will be less responsive (or "interactive") than its 100-data-point counterpart.

Step 4: Statistical Testing

As mentioned, data are generated serially from a (SLAM II) simulation program. Because of the possible presence of serial correlation, the "batch-means method" (see, for example, Law and Kelton [46]) is used to process data generated from the experiments. The statistical testing procedure is based on the following rationale.

First note that every job for which data are collected is fifty jobs away from other such jobs on average. In our implementation of the batch-means method, one hundred of these "spaced apart" jobs are collected, from which a batch is made, and a mean then calculated. We calculate 230 such batch means for each case in each experiment. To a first approximation, each batch mean can be assumed to be independent from every other batch mean, and all batch means can be reckoned to be normally distributed (which implies the batch means are i.i.d.). Independence occurs because each batch mean is, on average, 50*100 observations apart, thereby effectively eliminating all serial correlation; normality is obtained by virtue of the central limit theorem.

Now consider the first experiment. The basic test in this experiment is to see whether interactive prediction differs costwise from non-interactive prediction. This test is repeated twenty times: once for each of the four cost scenarios, and once for each of the five prediction techniques (neural networks, regression -- linear and nonlinear in variables, and two practitioner-based approaches). Note that each of the twenty cells has 230 observations (i.e., batch means).

Since there is sufficient power due to the large sample size and since batch means are normally distributed, the paired-t test is used. That is, for each technique and cost function tested, the difference is formed between the non-interactive and the interactive costs. These differences will be i.i.d. random variables which are also normally distributed. Note that with this test there is no requirement of equal variances (as would be required with ANOVA, for instance), since there is only one probability distribution in the paired-t test -- the distribution of the paired differences. Finally note that the proper test is 2-sided (i.e., a 2-sided, paired-t test), as we cannot postulate *a priori* whether interactive training or non-interactive training is less costly.

In the second experiment, the better linear regression case (i.e., interactive or non-interactive) as determined from the first experiment, is compared with the better nonlinear regression case, also as discovered from the initial experiment. The two practitioner-based techniques are dropped from analysis in experiment two for ease of exposition as they are expected *a priori* to be dominated by the other two methods.

With reference to the second experiment, observe that in order to compare the better linear case against the better nonlinear case, there are altogether four possible test comparisons that might have to be made. However, it is not known until the first experiment is over which of the four will be needed in the second experiment. Hence the second experiment constitutes a so-called *a posteriori* (or *post hoc*) test. For this test, because multiple comparisons are implicitly made with the same data, the experiment-wise error rate must be controlled. To do this, the conservative Bonferroni inequality is employed to adjust the α=0.01 significance level of the test appropriately. Once again, 2-sided, paired-t tests are in order.

In the third experiment, the better neural network case (either interactive or non-interactive) as determined from the first experiment is compared with the best regression case, as discovered from the first two experiments. Once again, this experiment constitutes an *a posteriori* test. In this instance there are eight possible multiple comparisons, since before any experimentation it is not known which cases will be preferred: four possible cases comparing non-interactive neural networks against the linear/nonlinear, interactive/non-interactive regression possibilities, and four more potential tests comparing interactive neural networks against the same four regression possibilities. Once again, the experimentwise error rate is controlled by using Bonferroni's inequality, and 2-sided, paired-t tests at a significance level of α=0.01 are used.

Results

In summary, the procedure described above was followed to generate 2,300 data points from the shop shown in Figure 5-3 for every PREDICTOR-technique and cost-scenario combination. The shop assumed immediate release and EDD dispatching. In the non-interactive case, the PREDICTORS for each technique were not updated at all; only initially determined prediction parameters were utilized. In the interactive case, each technique's PREDICTOR was updated as every job left the shop.

In the first experiment the 2-sided paired-t test was used to test the hypothesis that there is no difference costwise between non-interactive and interactive prediction. Results from this test are shown in Table 5-2. Each number in that table is the average of 230 batch costs, whereby each batch cost is the sum of earliness and tardiness costs (evaluated with the appropriate cost function) over the 100 jobs representing a batch. (As the batch size was 500 in the nonlinear, interactive regression case, the respective batch costs were scaled to make them comparable).

This table indicates that for practitioner-based rules (WIQ and JIQ), interactive training is always preferred in a significant manner. This suggests that failure to use an interactive approach is costly to practitioners, regardless of which cost scenario is present in their firms. Table 5-2 also leads to the same conclusion for the linear regression case: interactive training is less costly in a statistically significant manner irrespective of the type of earliness and tardiness costs.

Examination of Table 5-2 indicates that interactive training is also less costly for the nonlinear regression case, but that this is valid statistically only for symmetric cost functions. The converse is true for the neural network case: interactive training is

Table 5-2: Interactive vs. non-interactive training

Symmetric cost functions

Cost function: linear 1:1

training	NN	REG (nonlin.)	REG (lin.)	WIQ	JIQ
non-interactive	1,058	1,016	1,400	1,470	1,418
interactive	1,078	856	982	997	1,033
2-sided paired-t test	insignificant	p < 0.001	p < 0.001	p < 0.001	p < 0.001

Cost function: quadratic

training	NN	REG (nonlin.)	REG (lin.)	WIQ	JIQ
non-interactive	9,289	9,302	15,728	17,354	15,886
interactive	9,546	6,587	8,030	8,303	8,926
2-sided paired-t test	insignificant	p < 0.001	p < 0.001	p < 0.001	p < 0.001

Asymmetric cost functions

Cost function: linear 1:10

training	NN	REG (nonlin.)	REG (lin.)	WIQ	JIQ
non-interactive	2,750	4,874	6,665	7,252	6,384
interactive	2,442	4,739	5,252	5,305	5,507
2-sided paired-t test	p < 0.001	insignificant	p < 0.001	p < 0.001	p < 0.001

Cost function: linear-quadratic

training	NN	REG (nonlin.)	REG (lin.)	WIQ	JIQ
non-interactive	3,157	4,062	8,268	9,836	7,980
interactive	1,884	3,698	4,354	4,591	4,869
2-sided paired-t test	p < 0.001	p < 0.05	p < 0.001	p < 0.001	p < 0.001

preferred significantly for asymmetric cost functions. At this point, we make no strong conclusions regarding the benefits of interactive training for either neural networks or nonlinear regression; the preferred training appears to be case sensitive. As a final note to the first experiment, observe that all research reported in this chapter deals only with processes that are stationary (over time). We postulate that nonstationary processes whose job-service-time means, for example, vary widely with time would be expected to be predicted more accurately by interactive than non-interactive training, because the former could track such changes, whereas the latter could not. But this is an item for future research.

In the second experiment, the better linear regression case (i.e., either interactive or non-interactive) as determined from the first experiment was compared statistically with the better nonlinear regression case; the two practitioner-based rules were ignored. That is, the underlined numbers in the second technique's column in Table 5-2 were compared with the underlined numbers in the third technique's column. As explained, since this test is an *a posteriori* test, the experimentwise error rate was controlled using Bonferroni's inequality.

Results shown in Table 5-3 indicate that it is always significantly advantageous to predict due dates with second order variables allowed in the regression model. This result is not surprising, but it must be recalled that almost no theoretical or practitioner-based research reported in the literature reports nonlinear models. We suggest that if practice is reflective of the literature, very large penalties are being incurred.

Table 5-3: Linear vs. nonlinear regression

Symmetric cost functions

Cost function: linear 1:1

training	REG (nonlin.)	REG (lin.)	2-sided paired-t test
better of interactive or non-interactive	856	982	significant

Cost function: quadratic

training	REG (nonlin.)	REG (lin.)	2-sided paired-t test
better of interactive or non-interactive	6,587	8,030	significant

Asymmetric cost functions

Cost function: linear 1:10

training	REG (nonlin.)	REG (lin.)	2-sided paired-t test
better of interactive or non-interactive	4,739	5,252	significant

Cost function: linear-quadratic

training	REG (nonlin.)	REG (lin.)	2-sided paired-t test
better of interactive or non-interactive	3,698	4,354	significant

The final (third) experiment considered in this chapter was to evaluate the cost implications of not using neural networks to predict due dates. That is, a comparison was made between the better neural network approach (i.e., interactive or non-interactive training) and the best regression technique (linear/nonlinear, interactive/non-interactive). Again, Bonferroni's inequality was used to control for the eight possible comparisons in this *post hoc* test.

Table 5-4 shows the results of this test, all of which are statistically significant at α=0.01. The conclusion is that interactively-trained nonlinear-variables regression should be used when symmetric cost functions prevail, and interactively-trained neural networks should be invoked when asymmetric cost functions obtain. Basically, nonlinear regression outperforms neural networks in due-date prediction, except when the latter's advantage in handling asymmetric costs takes over. To see why this is the case, we now show how interactively trained neural networks' due-date predictions track their corresponding actual flow times.

Two cases of due-date prediction and actual flow times are presented in Figure 5-5. In Figure 5-5a, a symmetric quadratic cost function case, actual and predicted flow times through the shop for one set of serially generated batches (the abscissa) is shown. The tracking is impressive; the neural network's interactive-training PREDICTOR is working well.

In Figure 5-5b, the asymmetric linear 1:10 cost case is plotted. Recall that in that case the cost of tardiness varies linearly with the amount of tardiness, and, for an equal amount of tardiness, is ten times the cost of earliness. The initial impression from Figure 5-5b may be that the neural network interactive predictor is performing

Table 5-4: Neural networks vs. best OLS regression

Symmetric cost functions

Cost function: linear 1:1

training	NN	Best REG	2-sided paired-t test
better of interactive or non-interactive	1,058	856	significant

Cost function: quadratic

training	NN	Best REG	2-sided paired-t test
better of interactive or non-interactive	9,289	6,587	significant

Asymmetric cost functions

Cost function: linear 1:10

training	NN	Best REG	2-sided paired-t test
better of interactive or non-interactive	2,442	4,739	significant

Cost function: linear-quadratic

training	NN	Best REG	2-sided paired-t test
better of interactive or non-interactive	1,884	3,698	significant

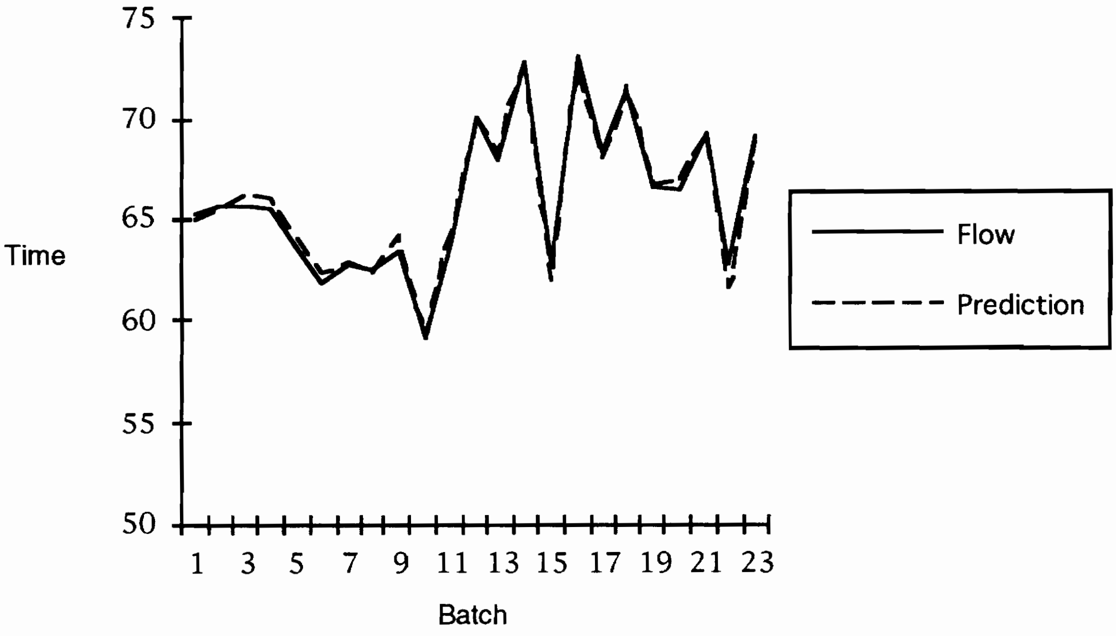


Figure 5-5a: Symmetric cost function

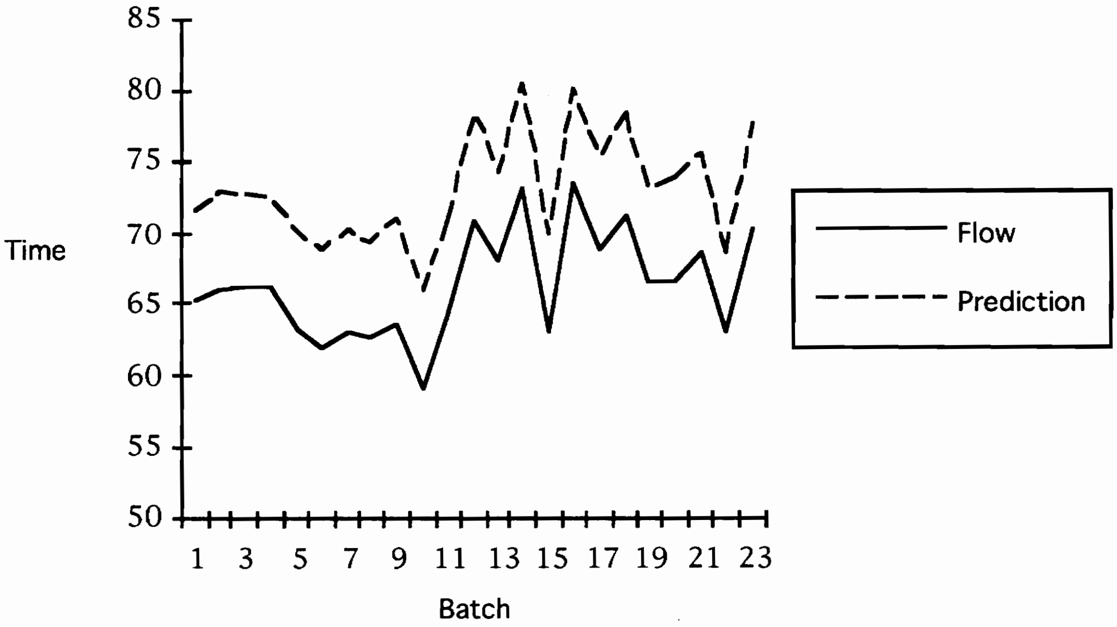


Figure 5-5b: Asymmetric cost function

Figure 5-5: Neural network flow tracking

poorly, consistently overestimating the flow time. However, the neural network is not performing poorly at all. Rather, it consistently is biased high, because the cost of tardiness is so high relative to the cost of earliness, that the economically best decision is to consistently set the predicted due date late. The cost data in the tables support this hypothesis. The regression PREDICTOR estimates (not shown) are not able to compensate like this. Therefore, the neural network interactive PREDICTOR is preferred for asymmetric cost cases.

An additional informal experiment, not previously discussed, returns to the basic question of whether due-date sequencing itself is warranted, regardless of whether it utilized an interactive predictor or not. In a comparison with the non-due-date-based rule SPT, it was found for all four cost functions and all five techniques reported here that SPT was significantly more costly than due-date-based sequencing for either the interactive or the non-interactive prediction approach. This provides additional justification for this chapter, where we are searching for the best due-date prediction technique.

Conclusions And Future Work

In this chapter of the dissertation we have studied the effect of interactive prediction on different techniques for the due-date setting problem. In so doing, this part of the research has provided a coherent integration of the first and third stages of the production control system, namely, due-date prediction and shop-floor control. Moreover, a methodology was developed and demonstrated to be effective in performing this interactive integration.

Results were generated using four cost-based criteria. It is believed that these criteria, two symmetric and two asymmetric, are the proper way to evaluate alternatives, as implicit assumptions (about the cost of earliness and tardiness) buried in other criteria are explicitly surfaced. It was found that for symmetric cost functions, interactively-trained nonlinear regression models are significantly preferred. Moreover, it was discovered that for asymmetric cost functions, interactively-trained neural networks are the methodology of choice.

Although the results generated may safely be extended only to shops like that studied in this chapter, the possibility that similar results might be found for other shops points out the need for further research on additional shop environments. We note that the primary conclusions reached here that either neural networks or nonlinear regression models should be used with interactive training do not appear to "square" with practice, if the current literature is any indication. To the best of our knowledge neither neural networks, nonlinear regression models, nor interactive training approaches are utilized widely in practice. We believe the cost implications to be profound.

In this regard, it should also be noted that, again from a cost perspective, due-date-based sequencing was found to be superior to SPT dispatching. This is consonant with other findings in the literature (e.g., Ragatz and Mabert [62]) using other criteria. This result adds further credence to this and similar research, where the purpose is to ascertain the best way to practice due-date-based dispatching.

As a matter of further interest to the reader, we furnish Table 5-5, the results of an additional study we performed on the effect of pre-training batch size. That table indicates that the larger the pre-training batch size (3,500 pre-training data points in the first results' column versus 500 in the second column), the lower the cost.

Table 5-5: 3,500 vs. 500 neural network pre-training jobs

Symmetric cost functions

Cost function: linear 1:1

training	NN 3500	NN 500	2-sided paired-t test
non-interactive	974	1058	p < 0.001
interactive	1,029	1,078	p < 0.001

Cost function: quadratic

training	NN 3500	NN 500	2-sided paired-t test
non-interactive	8,568	9,289	p < 0.05
interactive	9,376	9,546	insignificant

Asymmetric cost functions

Cost function: linear 1:10

training	NN 3500	NN 500	2-sided paired-t test
non-interactive	2,234	2,750	p < 0.001
interactive	2,289	2,442	insignificant

Cost function: linear-quadratic

training	NN 3500	NN 500	2-sided paired-t test
non-interactive	2,491	3,157	p < 0.001
interactive	1,866	1,884	insignificant

Finally, in fairness, it must be pointed out that the scope of this research is, again, limited in that implementation issues have been ignored. Although there is no reason to doubt that the results here can be applied in the "real" world, further research is needed to verify this. In addition, further work determining earliness and tardiness cost functions based on practitioner data would be worthwhile to see if the functional forms used here are realistic. Also, inclusion of the second stage of the production control system, order release, in an integrated, cost-based manner is a topic for further study.

Chapter 6

CONCLUSIONS, CONTRIBUTIONS, AND FUTURE WORK

This dissertation has explored prediction and control issues in a production setting. In particular, a fairly new artificial intelligence technique, neural networks, was examined within this context.

This research had three objectives. The primary purpose was to see whether neural networks hold any promise in the area of due-date prediction, the first stage of shop control. A second objective was to introduce cost measures into the prediction scheme of due-date assignment techniques as a criterion of comparison. The third objective was to study the interaction between the two main stages of shop control, due-date prediction and shop floor dispatching with both neural network and regression predictors.

With respect to the first objective, the issue of neural network promise in due-date predictions, a review of the relevant literature revealed a lack of sophisticated due-date prediction techniques. This was found to be the case, in spite of the need for improved production control due to increasing competitive pressures. In chapter 3 of this dissertation, two sophisticated due-date prediction techniques previously not studied, neural networks and nonlinear regression were examined . Both methods indicated improvement over the literature, thereby providing a positive finding with respect to the first objective. In particular, it was found that in two of three shops analyzed, the neural network outperformed conventional statistical methods, while doing no worse on the third. In the comparison of neural networks and regression, two additional conclusions were obtained. First, it was discovered that inclusion of additional explanatory variables does improve the performance of the neural network, while doing nothing to help least squares regression. This adds evidence to the postulate advanced by Hecht-Nielsen [31] that neural networks are significantly better than regression in higher dimensional input spaces. Second, it was observed that multicollinearity has a smaller deleterious effect on neural network prediction than on regression due dates.

The second objective, introducing cost measures directly into the prediction technique, was motivated by the observation that practitioners and researchers alike generally assume cost-of-earliness and cost-of-tardiness functions with little resemblance to true costs. A methodology was presented for explicitly considering costs in the due-date prediction problem. With respect to a best approach for due-date prediction, OLS regression, linear and quadratic programming, and neural networks were evaluated. It was found that: (1) ignoring cost-based predictions can be very costly; (2) simpler regression-based rules cited in the literature are very poor cost performers; and (3) if the form of the earliness cost function differs from that of the tardiness cost function (which is believed here to be the most realistic scenario), neural networks are statistically superior performers. Furthermore, neural networks can be used for essentially any differentiable cost function desired, whereas other methodologies are significantly more restricted. Finally, neural networks indicated a certain robustness as they performed well on all cost functions studied.

The third objective, studying the interaction between due-date prediction and shop-floor dispatching, was initiated by the premise (supported in part by the literature and in part by practice) that due-date based dispatching rules are less costly than non-due-date based rules. Therefore, integrating these two stages of shop control had the potential for pleasant economic ramifications as well as theoretical benefits. Such, in fact, was found to be the case in chapter 5. In that chapter a due-date-based dispatching rule was considered which introduced interaction between the due-date prediction and the dispatching decision. Interactive due-date prediction procedures were outlined for both OLS regression as well as neural networks. From a cost-based analysis of the different techniques it was found for the scenarios considered that: (1) due-date-based dispatching always resulted in lower costs than SPT (i.e., non-due-date-based) dispatching; (2) interactive training always significantly improved the performance of the conventional regression techniques; (3) including nonlinear independent variables in the regression model was always significantly preferred over the use of only linear variables; (4) interactive, nonlinear-variables regression was the best technique for symmetric cost functions; and (5) neural networks proved statistically superior for asymmetric cost functions.

It is noted, therefore, that all three objectives set forth in this dissertation have been met.

Attention is now focused on promising areas for future work. From a mainly theoretical perspective, other neural network paradigms as well as more sophisticated nonlinear regression models should be evaluated. In this research the only neural network paradigm used was backpropagation. Moreover, only minimal experimentation with respect to squashing functions, number of nodes in the input or hidden layer, learning and momentum coefficient, etc., was conducted. This was because the neural networks utilized, simple as they were, in general met all research objectives. Nonetheless, since additional savings are theoretically possible, further research should be carried out. A particular example believed to hold much promise (because success has been experienced with such transformations by the author in other research endeavors) would be to transform the input data into orthogonal polynomial representations of those data. (See, for example, Pao [58], for elaboration on this approach.)

A more pragmatic area for future work encompasses practitioner issues. All analyses in this dissertation were based on data generated by the simulation of hypothetical shops. Moreover, relatively very few shops were studied. It is therefore necessary that the findings reported here be tested on a wide variety of "real" shops with "real" data. No problem is anticipated using neural networks in "real" shops, but until it is done, the implementation issue remains alive. Moreover, the collection of data in an effort to determine realistic earliness and tardiness cost functions is needed. It is suspected that such research will confirm the suspicion advanced in this dissertation of asymmetric real-world cost functions, thereby increasing the attractiveness of neural networks as due-date setters.

A final area of future work mentioned here is to continue with the basic theme of this dissertation, namely providing effective and efficient analysis of integrated shop control. The different procedures and techniques motivated and developed in this work should further be evaluated on additional shop-floor dispatching rules. Then, although integration of two of the three areas of shop control has been examined in this research, the influence of the third area, order release, should also be studied. Finally, it is suggested that a cost-based, totally-integrated shop control system utilizing sophisticated techniques would provide a challenging next step beyond this research.

References

1. Ackerman, S. S., "Even Flow, A Scheduling Method for Reducing Lateness in Job Shops," *Management Technology*, 3 (1963), 1, 20-32.

2. Akers, L., and Walker, M., "A limited-interconnect synthetic neural IC," *Proceedings of the IEEE International Conference on Neural Networks: Vol. II*, (1988), 151-158.

3. Anderson, J. A., Silverstein, J. W., Ritz, S. A., and Jones, R. S., "Distinctive features, categorial perception, and probability learning: Some applications of a neural model," *Psychological Review*, 84 (1977), 413-451.

4. Baker, Kenneth R., "Sequencing Rules and Due-Date Assignment in a Job Shop," *Management Science*, 30 (1984), 1093-1104.

5. Baker, Kenneth R., "The Effects of Input Control In a Simple Scheduling Model," *Journal of Operations Management*, 4, 2 (1984), 99-112.

6. Baker, K. R., and Bertrand, J. W. M., "A Comparison of Due-Date Selection Rules," *IIE Transactions*, 13 (1981), 123-131.

7. Bechte, W., "Controlling Manufacturing Lead Time and Work-in-Process Inventory by Means of Load-Oriented Order Release," *Proceedings, APICS Conferences*, (1982), 67-92.

8. Bertrand, J. W. M., "The Use of Workload Information to Control Job Lateness in Controlled and Uncontrolled Release Production Systems," *Journal of Operations Management*, 3 (1983), 2, 67-78.

9. Blackstone, J. H. Jr., Phillips, D. T., and Hogg, G. L., "A State of the Art Survey of Sequencing Rules for Manufacturing Job Shop Operations," *International Journal of Production Research*, 20 (1982), 27-45.

10. Bobrowski, P. M., and Park, P. S., "Work Release Strategies in a Dual Resource Constraint Job Shop," *OMEGA*, 17 (1989), 2, 177-188.

11. Bookbinder, J. H., and Noor, A. I., "Setting Job-Shop Due-Dates with Service-Level Constraints," *Journal of the Operational Research Society*, 36, 11 (1985), 1017-1026.

12. Burr, D., "A neural network digit recognizer," *Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics*, (1986), 1621-1625.

13. Cheng, T. C. E., "Integration of Priority Dispatching and Due Date Assignment in a Job Shop," *International Journal of Systems Science*, 19 (1988), 9, 1813-1825.

14. Cheng, T. C. E., and Gupta, M. C., "Survey of Scheduling Research Involving Due Date Determination Decisions," *European Journal of Operational Research*, 38 (1989), 156-166.

15. Christy, D. P., and Kanet, J. J., "Manufacturing Systems with Forbidden Early Shipment: Implications for Choice of Scheduling Rules," *International Journal of Production Research*, 1, 28 (1990), 91-100.

16. Conway, R. W., "Priority Dispatching and Job Lateness in a Job Shop," *Journal of Industrial Engineering*, 16 (1965), 228-237.

17. Cottrell, G., Munro, P., and Zipser, D., "Image compression by backpropagation," *Technical Report 8702*, University of California at San Diego, Institute for Cognitive Science, 1987.

18. Day, J. E., and Hottenstein, M. P., "Review of Sequencing Research," *Naval Research Logistics Quarterly*, 17 (1970), 1, 11-39.

19. Eilon, S., "Production Scheduling," in K. B. Haley (ed.), *OR '78*, North-Holland, Amsterdam, 1978, 237-266.

20. Eilon, S., and Chowdhury, I. G., "Due Date in Job Shop Scheduling," *International Journal of Production Research*, 14 (1976), 223-238.

21. Elmaghraby, S. E., "The Machine Sequencing Problem Review and Extensions," *Naval Research Logistics Quarterly*, 15 (1960), 2, 205-232.

22. Elvers, D. A., "Job Shop Dispatching Rules Using Various Delivery Date Setting criteria," *Production and Inventory Management*, 14 (1973), 62-70.

23. Fiesler, E., Chandry, A., and Caulfield, H., "Weight descretization in backward error propagation neural networks," *Neural Networks Supplement: INNS Abstracts*, (1988), 1, 380.

24. Fry, T. D., "Controlling Input: The Real Key to Shorter Lead-Times," *The International Journal of Logistics Management,* 1, 1 (1990), 7-12.

25. Fry, T. D., and Smith, A. E., "Input/Output Controls as ABC Tool," *Production and Inventory Management Journal,* 28, 3 (1987), 50-52.

26. Graves, S. C., "Review of Production Scheduling," *Operations Research*, 29 (1981), 3, 646-675.

27. Grossberg, S., "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors," *Biological Cypernetics*, 23 (1976), 121-134.

28. Gupta, S. K., and Kyparisis, J., "Single Machine Scheduling Research," *OMEGA*, 15 (1987), 207-227.

29. Hebb, D., *The Organization of Behavior,* Wiley, New York, 1949.

30. Hecht-Nielsen, R., "Counterpropagation networks," *Proceedings of the IEEE First International Conference on Neural Networks: Vol. II*, (1987), 19-32.

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

32. Hinton, G. E., Sejnowski, T. J., and Ackley, D. H., "Boltzmann machines: Constraint satisfaction networks that learn," *Technical Report CMU-CS-84-119*, Carnegie-Mellon University, Department of Computer Science, PA, 1984.

33. Hinton, G. E., and Touretzky, D. S., "Neural Networks for Real-World Problems," *Tutorial SP2,* The Ninth National Conference on Artificial Intelligence, Anaheim, CA, 1991.

34. Hopfield, J. J., "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, 79 (1982), 2554-2558.

35. Hopfield, J. J., "Neurons with graded response have collective computational abilities," *Proceedings of the National Academy of Sciences*, 81 (1984), 3088-3092.

36. "IMSL User's Manual," *IMSL STAT/Library, Vol.1,* Version 1.1, 1989.

37. Irastorza, J. C., and Deanne, R. H., "A Loading and Balancing Methodology for Job Shop control," *AIEE Transactions*, 6 (1974), 302-307.

38. Jones, C. H., "An Economic Evaluation of Job Shop Dispatching Rules," *Management Science*, 20 (1973), 293-303.

39. Kanet, John J., "Load-Limited Order Release in Job-Shop Scheduling Systems," *Journal of Operations Management,* 7, 3 (1988), 44-58.

40. Kanet, John J., "On Anomalies in Dynamic Ratio Type Scheduling Rules: A Clarifying Analysis," *Management Science*, 28 (1982), 1337-1341.

41. Kanet, J. J., and Christy, D. P., "Manufacturing Systems with Forbidden Early Shipment: Implications for Setting Manufacturing Lead Times," *International Journal of Production Research*, 5, 27 (1989), 783-792.

42. Klimasauskas, Casimir, Guiver, John, and Pelton, G., *NeuralWorks Professional II and NeuralWorks Explorer, Volume I*, Neural Computing, NeuralWare, Inc., Pittsburg, PA, 1989.

43. Kolmogorow, A. N., "On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition" [in Russian], *Dokl. Akad. Nauk. USSR,* 114 (1957), 953-956.

44. Kosko, B., "Bidirectional associative memories," *IEEE Transactions on Systems, Man, and Cybernetics*, 18 (1988), 1, 49-60.

45. Lapedes, A., and Farber, R., "Nonlinear signal processing using neural networks: Prediction and modeling," *Proceedings of the IEEE,* in review, (1988).

46. Law, Averill M., and Kelton, W. David, *Simulation Modeling and Analysis*, McGraw-Hill Book Co., New York, 1982.

47. Markham, Ina S., "An Exploration Of The Robustness Of Traditional Regression Analysis Versus Analysis Using Backpropagation Networks," *unpublished Ph.D. Dissertation,* Virginia Polytechnic Institute and State University, 1992.

48. McCulloch, W. S., and Pitts, W., "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, 5 (1943), 115-133.

49. Melnyk, S. A., and Ragatz, G. L., "Order Review/Release: Research Issues and Perspectives," *International Journal of Production Research*, 27 (1989), 7, 1081-1096.

50. Minsky, M., "Neural nets and the brain - Model problem," *Doctorial Dissertation,* Princeton University, 1954.

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

52. Moore, J. M., and Wilson, R. C., "A Review of Simulation Research in Job Shop Scheduling," *Production and Inventory Management*, 8 (1967), 1, 1-10.

53. Müller, B., and Reinhardt, J., *Neural Networks: An Introduction*, Springer-Verlag, New York, 1990.

54. Murtaugh, B. A., and Saunders, M. A., *MINOS 5.1 User's Guide, Technical Report SOL 83-20R*, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1987.

55. Myers, R. H., *Response Surface Methodology*, Allyn and Bacon, Boston, MA, 1971.

56. Neter, J., Wasserman, W., and Kutner, M. H., *Applied Linear Statistical Models,* (2nd Ed.), Irwin, IL, 1985.

57. Panwalker, S. S., and Iskander, W., "A Survey of Scheduling Rules," *Operations Research*, 25 (1977), 1, 45-61.

58. Pao, Y. H. *Adaptive Pattern Recognition and Neural Networks, Reading,* Addison-Wesley Publishing Company, Inc., MA, 1989.

59. Pritsker, A. Alan B., Sigal, C. Elliott, and Hammesfahr, R. D. Jack, *SLAM II: Network Models for Decision Support*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1989.

60. Ragatz, Gary L., and Mabert, Vincent A., "A Framework for the Study of Due Date Management in Job Shop," *International Journal of Production Research*, 22 (1984), 685-695.

61. Ragatz, Gary L., and Mabert, Vincent A., "An Evaluation of Order Release Mechanisms in a Job-Shop Environment," *Decision Sciences*, 19 (1988), 167-189.

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

63. Rakes, T. R., Rees, L. P., Siochi, F. C., and Wray, B. A., "Estimating the Number of Kanbans Using Neural Networks," *working paper,* Virginia Polytechnic Institute and State University, 1990.

64. Ramamoorthy, P., Gouird, G., and Iyer, V., "Signal modeling and prediction using neural networks," *Neural Networks Supplement: INNS Abstracts*, 1, (1988), 461.

65. Ransil, P., and Siegel, K., "Control of deformable mirror using an adaptive network," *Neural Networks Supplement: INNS Abstracts*, 1, (1988), 402.

66. Rosenblatt, F., *Principles of Neurodynamics*, Spartan Books, 1961.

67. Rosenblatt, F., "The perceptron: A probabalistic model for information storage and organization in the brain," *Psychological Review*, 65 (1958), 386-408.

68. Rumelhart, D. E., and McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, I & II*, MIT Press, Cambridge, MA, 1986.

69. Rumelhart, D. E., and Hinton. G. E., and Williams, R. J., "Learning internal representations by error propagation," in Rumelhart, D. E., and McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, I & II*, MIT Press, Cambridge, MA, 1986, 318-362.

70. SAS Institute Inc., *SAS User's Guide: Statistics, Version 5 Edition*, Cary, NC, SAS Institute Inc., 1985.

71. Scudder, G. D., and Hofman, T. R., "The Use of Value-Based Priorities when Early Shipments are Forbidden," *International Journal of Production Research*, 27 (1989), 2, 353-369.

72. Sejnowski, T., "Learned classification of sonar targets using a massively parallel network," *Workshop on Neural Network Devices and Applications*, Jet Propulsion Laboratory, (1987), 224-237.

73. Sejnowski, T., and Rosenberg, C., "NETtalk: A parallel network that learns to read aloud," *Technical Report JHU-EECS-86-01,* John Hopkins University, Department of Electrical Engineering and Computer Science, 1986.

74. Simpson, P. K., *Artificial Neural Systems,* Pergamon Press, New York, 1990.

75. Smith, C. H., and Gee, E. S., "On the Benefits of Using Global Variables in Setting Due Dates," *Proceedings of the Twentieth Annual Meeting, Southeast Region of the Decision Sciences Institute*, Columbia, SC, 1990, 232-234.

76. Smith, M. L., and Seidmann, A., "Due Date Selection Procedures for Job Shop Simulation," *Computers and Industrial Engineering*, 7 (1983), 199-207.

77. Tolat, V., and Widrow, B., "An adaptive 'broom balancer' with visual inputs," *Proceedings of the IEEE International Conference on Neural Networks: Vol II*, 1988, 362.

78. Wasserman, P. D., *Neural Computing,* Van Nostrand Reinhold, New York, 1989.

79. Weeks, J. K., "A Simulation Study of Predictable Due-Dates," *Management Science*, 25 (1979), 363-373.

80. Weeks, J. K., and Fryer, J. S., "A Simulation Study of Operating Policies in a Hypothetical Dual-Constrained Job Shop," *Management Science*, 22 (1976), 1362-1371.

81. Wein, Lawrence M., "Due-Date Setting and Priority Sequencing in a Multiclass M/G/1 Queue," *Management Science*, 37, 7 (1991), 834-850.

82. Widrow, B., "Generalization and Information storage in networks of ADALINE neurons," in Yovitts, G. T. [ed.], *Self-Organizing Systems*, Spartan Books, Washington DC, 1962.

83. Widrow, B., and Hoff, M. E., "Adaptive switching circuits," *1960 IRE WESCON Convention Record*, New York, 1960, 96-104.

84. Wight, O. W., "Input/Output Control: A Real Handle on Lead Time," *Production and Inventory Management*, 11 (1970), 3, 9-31.

85. Wray, Barry A., "Prediction and Control in a Just-In-Time Environment Using Neural Networks," *unpublished Ph.D. Dissertation,* Virginia Polytechnic Institute and State University, 1992.

Vita

Lars Wiegmann was born on July 18, 1964, in Hamburg, Germany. From 1970 until 1980 he attended first a primary and then a humanistic grammar school in Hamburg. He spent Grade 11 at a boarding school in Southport, Australia, before graduating in 1983 with the Abitur from the Schule Schloß Salem in Baden-Württemberg, Germany.

From 1983 to 1985 Mr. Wiegmann was employed by the Vereins- und Westbank AG, and then until 1986 by the Angermann Consult GmbH, both in Hamburg. During those years he also attended the Wirtschaftsakademie Hamburg from which he graduated as a Business Assistant and as a Business Administrator in 1985 and 1986 respectively. His performance in this three-year training program earned him a scholarship from the Übersee-Club, Hamburg. For the next two years from 1986 to 1988 he studied Industrial Engineering at the Universität Karlsruhe, completing his Vordiplom in 1987.

In 1988 Mr. Wiegmann came as a Fulbright scholar to Virginia Tech, where he completed his MBA in 1989. Since then he has been a member of the Ph.D. program in the Department of Management Science, serving as a Graduate Assistant, and working on his dissertation since August 1991.

Lars Wiegmann